

①

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 10-116295

(43)Date of publication of application : 06.05.1998

(51)Int.Cl. G06F 17/30
G06F 13/00

(21)Application number : 09-190827

(71)Applicant : RICOH CO LTD

(22)Date of filing : 16.07.1997

(72)Inventor : STEPHEN R SABIKKI
GREGG WOLF

(30)Priority

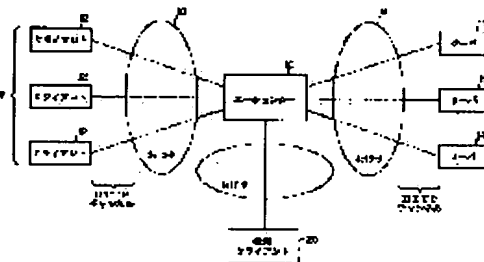
Priority number : 96 718858 Priority date : 24.09.1996 Priority country : US

(54) DOCUMENT AGENCY SYSTEM

(57)Abstract:

PROBLEM TO BE SOLVED: To make the execution of a general agent allowable at a position specified by a user by sending an answer to a client as indicated by a consistent agent, an agent which gives satisfaction, or a transaction in a transaction sequence.

SOLUTION: A user can access an agency 10 directly, but typically takes some kind of action at a Web client 12 to access the agency 10 so as to access a Web server 14. This agency 10 is transparent to the Web client 12 in many cases and the Web client 12 and Web server 14 are so programmed as to communicate with each other by using a hypertext document transmission program like HTTP over a channel 16. Thus, an answer is sent to the client as indicated by the consistent agent, agent which gives satisfaction, or transaction in the transaction sequence.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-116295

(43) 公開日 平成10年(1998) 5月6日

(51) Int.Cl.⁶

G 0 6 F 17/30
13/00

識別記号

3 5 1

F I

G 0 6 F 15/40
13/00
15/40

3 8 0 Z
3 5 1 G
3 1 0 F
3 1 0 C

審査請求 未請求 請求項の数17 O L (全 24 頁)

(21) 出願番号 特願平9-190827

(22) 出願日 平成9年(1997) 7月16日

(31) 優先権主張番号 08/718858

(32) 優先日 1996年9月24日

(33) 優先権主張国 米国 (U S)

(71) 出願人 000006747

株式会社リコー

東京都大田区中馬込1丁目3番6号

(72) 発明者 ステフェン アール サビッキー

アメリカ合衆国 94025, カリフォルニア

メンロパーク 115, スート アールデ

イ サンド ヒル 2882 リコーコーポレ
ーション内

(72) 発明者 グレグ ウォルフ

アメリカ合衆国 94025, カリフォルニア

メンロパーク 115, スート アールデ

イ サンド ヒル 2882 リコーコーポレ
ーション内

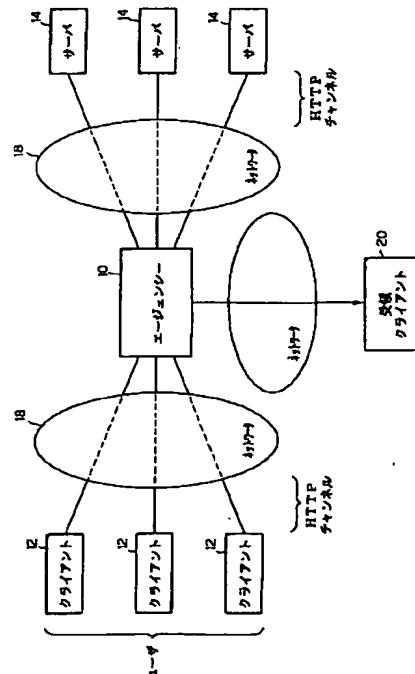
(74) 代理人 弁理士 高野 明近 (外1名)

(54) 【発明の名称】 ドキュメントエージェンシーシステム

(57) 【要約】

【課題】 WWWのようなドキュメント指向のフレームワークで動作し、使用されるサーバ又はクライアントとは無関係にユーザによって特定された位置でエージェントを実行可能にする。

【解決手段】 ドキュメント指向計算システムであるWebエージェンシーは、要求をWebサーバに送るに先立って、Webクライアントからの要求を変換し、かつ、Webエージェンシーに接続された種々のWebクライアントのユーザについての状態情報を蓄積するため、WebクライアントとWebサーバの間に挿入されている。Webエージェンシーは、Webクライアント及びWebサーバに透過可能でかつそれらが独立している。Webエージェンシーは周辺エージェンシー、個人サービスエージェンシー又はドキュメントサーバとして使用できる。



【特許請求の範囲】

【請求項1】 クライアントからドキュメント要求を受け取り；サーバから応答を受け取り、トランザクション列中にトランザクションを生成し；トランザクション列中の各トランザクションを、トランザクションから情報を抽出し；トランザクションを変更し又はトランザクション列に付加すべき新規なトランザクションを生成する、一又はそれを越える整合エージェントに整合させ；トランザクションによって特定されるメソッド又は満足を与えるエージェントを用いて各トランザクションを満足させ、満足を与えるエージェント又はメソッドはクライアント及びサーバとは独立しており；整合エージェント、満足を与えるエージェント又はトランザクション列中のトランザクションによって指示されるように、サーバにドキュメント要求を送り；かつ、整合エージェント、満足を与えるエージェント又はトランザクション列中のトランザクションによって指示されるように、クライアントに応答を送る；各ステップからなるドキュメント検索及び操作方法。

【請求項2】 少なくとも整合のステップが特定のユーザのために保持された状態に応じて実施されることを特徴とする、特定のユーザのための状態を保持するステップからさらに成っている請求項1の方法。

【請求項3】 少なくとも整合のステップが周辺機器のために保持された状態に応じて実施されることを特徴とする、周辺機器のための状態を保持するステップからさらに成っている請求項1の方法。

【請求項4】 クライアントがドキュメント転送プロトコルによってサーバからドキュメントを要求するネットワークにおける、クライアント及びサーバと独立にネットワークの機能性を拡張する方法であって、特定のユーザのための状態を蓄積することを含む、特定のユーザによる使用のための個人エージェントをカスタマイズし；特定のユーザからのドキュメント要求を受け入れ、その際、ドキュメント要求は指定されたサーバに向けられかつ要求するクライアントから開始されるものであって；特定のユーザと協働する個人エージェントにドキュメント要求を発送し；指定されたサーバにドキュメント要求を発送し；ドキュメント要求を反映させるため状態を更新し；ドキュメント要求で要求されたドキュメントを、個人エージェントのところで受け取り；個人エージェントの状態によって指示されるようにドキュメントを変換する；各ステップからなる方法。

【請求項5】 特定のユーザは個人エージェントを用いる個人のグループの一人である、請求項4の方法。

【請求項6】 ドキュメントを要求クライアントと異なるクライアントである受容クライアントへ発送するステップからなる、請求項4の方法。

【請求項7】 特定のユーザによって要求されたドキュメントの履歴を、個人エージェントによって、蓄積す

るステップからなる、請求項4の方法。

【請求項8】 エージェントが、転送のユニットがドキュメントであるハイパーテキスト転送プロトコルを用いるクライアント及びサーバと通信する、請求項4の方法。

【請求項9】 クライアント要求を周辺機器から提供するネットワークにおいて、ドキュメント転送プロトコルを用いてクライアントからドキュメント処理要求を送り；周辺エージェントのところでドキュメントを獲得し；周辺エージェントからの制御ドキュメントをクライアントに発送し、その際、制御ドキュメントはドキュメントの表示及びドキュメントに対して採る動作を特定する制御部を含み；周辺機器によるドキュメントの処理を開始するためクライアントからの制御命令を、ドキュメント転送プロトコルを用いて、クライアントから周辺エージェントに送るステップからなる、クライアントを周辺機器にインターフェースする方法。

【請求項10】 多数のエージェントの表示を蓄積するエージェント配列；トランザクション列；ドキュメント要求及びドキュメントをWebクライアント又はWebサーバから受け入れるためのプログラミング、及び、トランザクション列中に表示され、かつ、エージェント配列中の一又はそれを越えるエージェントによって動作されるトランザクションを生成するためのプログラミング含むハイパーテキスト転送プロトコルマネジャー；トランザクションをエージェントに整合させ、かつ、そのエージェントと整合するトランザクション列中のトランザクションを処理するために必要なとき、エージェント配列中のエージェントを活動させるためにエージェント配列及びトランザクション列に結合されるリゾルバ；からなる、クライアントとサーバの間で転送されるドキュメント及びドキュメント要求を変換するため、WebクライアントとWebサーバの間でネットワークに挿入されたWebエージェント。

【請求項11】 Webエージェントは周辺エージェントであり、周辺エージェントは、周辺機器によるドキュメント処理要求を仲介するため、周辺機器と一又はそれを越えるブラウザクライアント間に挿入され、周辺機器は周辺機器と対話するのに用いられる一又はそれを越えるブラウザクライアントから独立している、請求項10の装置。

【請求項12】 Webエージェントがドキュメントファイリングシステムへのアクセスを提供しかつ制御する請求項10の装置。

【請求項13】 Webエージェントは個人情報エージェントであり、個人情報エージェントは個人の状態を保持するため及び保持された個人の状態に応じてドキュメントとドキュメント要求を変更するため、ユーザのクライアントとネットワーク間に挿入されたエージェントである、請求項10の装置。

【請求項14】 個人の状態が、個人情報エージェンシーのユーザによって動作される多数の独立したクライアントを包括する選択された資源の包括的個人ホトリストからなる、請求項13の装置。

【請求項15】 個人の状態が、個人情報エージェンシーのユーザによって使用される選択された資源の包括的な個人履歴リストからなっており、包括的な個人履歴リストは個人情報エージェンシーのユーザによって動作される多数の独立したクライアントを包括するものである、請求13の装置。

【請求項16】 ハイパーテキストの部分を実行可能なフォームにフォーマットし；リンクされた参照（link references）から機械読み取り可能なプリントシンボルを生成し；機械読み取り可能なプリントシンボルの表示をリンクされた参照用のアンカーの位置に対応した位置でハイパーテキストドキュメントのプリント可能なフォームに挿入し；かつ、ハイパーテキストドキュメントのプリント可能なフォームを実行可能なフォームをプリントする；各ステップからなる、ハイパーテキストドキュメントの一部を表すプリントされたドキュメント中にハイパーテキストリンクを保持する方法。

【請求項17】 機械読み取り可能なプリントシンボルがハイパーテキストドキュメントのプリント可能なフォーム中のアンカーの下にアンダーラインを形成する、請求項16の方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、情報の取り扱い及び処理分野に関するものである。より詳細には、一実施例において、本発明は、インターネットのようなネットワーク上で伝送されるドキュメントにおいて具体化される改善された情報処理を提供するものである。

【0002】

【従来の技術】 用語“インターネット”は最近広範な利用を勝ち得ているネットワークである世界的なインターネットワークを指している。

【0003】 任意の2台のコンピュータ（ノード）の通信は、両者がどこかでインターネットに接続されている限り、インターネット上で行うことができる。インターネット上でのデータ通信の最も一般的なプロトコルはTCP/IP（Transport Control Protocol / Internet Protocol）である。TCP/IPの上のレイヤー

（層）における多数の異なるトランスポート（transport）アプリケーションが知られている。FTP（File Transfer Protocol）は、ファイルを一方のマシンから別のマシンへ移動させるのに用いられるプロトコルである。その利用故に、典型的なFTPクライアント（client）は、ファイルのリストを表示して、ディレクトリー構造などを通過できるようにするファイルマネージャーに似たインターフェースを有している。SMTP（Simple

Mail Transfer Protocol）は、通常、ユーザの対話（interaction）なしで、その最終的な行き先の途中でホストマシン間でeメールを移動させるのに使用されるプロトコルである。HTTP（Hyper Text Transfer Protocol）は、殆どのドキュメントがワールドワイドWeb（即ち、“www”又は“Web”）ドキュメントである場合に、ノード間でドキュメントを伝送するのに使用されるプロトコルである。典型的には、HTTPはWeb（Web）サーバとWebブラウザクライアント（client）間で使われている。HTTPを用いて伝送されるドキュメントは、グラフィック要素及びフォーム（形式）のような、幾つかのコンポーネント（成分）から構成されている。Webブラウザはエンドユーザ（最終利用者）に見せる（ドキュメント、グラフィック又はビデオドキュメント）又は聞かせる（オーディオドキュメント）ために、ドキュメントを表示することができ、かつ、一方の参照ドキュメントから別のものにジャンプすることができるため、エンドユーザは、HTTP形態を用いて提供されるドキュメントがドキュメント網を形成していることを知る。従って、ワールドワイドWeb（“www”又は“Web”）は、HTTPを用いるインターネット上で利用可能な数千の項目によって収集されリンクされたドキュメントの集合体を指すのに用いられている。

【0004】 基本的には、HTTPはドキュメントトランスポートプロトコルである。現在のHTTPシステムにおいては、HTTPクライアント（必ずしもという訳ではないが、通常はブラウザである）は、HTTPのサーバに対してドキュメントを要求するためHTTP要求（リクエスト）を送り、（もし、そのクライアントがそのドキュメントを読む許可を得ていれば）そのドキュメントを返送する。もし、そのドキュメントが複合ドキュメント（a compound document）であれば、HTTPクライアントは埋め込まれたドキュメント（典型的にはグラフィックイメージである）への参照を見つけるために返送されたドキュメントを読む。Webサーバは、ドキュメントイメージを形成するのに十分なデータを受け取ると、そのドキュメントのイメージを表示する。Webの益々大きくなる人気は、その一端はドキュメントを見るのに利用者は単にドキュメントを要求すればよいという事実によるものである。これとは対照的に、FTPでは、ユーザは、ファイルリストからファイルを選び、そのファイルを伝送し、埋め込まれたファイルが必要か否かを決定するためにそのファイルを開き、リストからそれらのファイルを選択し、それらをダウンロードし、かつ、それらから最後にそのドキュメントを可視化しなければならない。

【0005】 応用例のあるものにおいては、既存のドキュメントを単に伝送すること以上のことが要求される。固定のドキュメントの代わりに、利用者はドキュメント

の要求が受け入れられた時に、動的に生成されるドキュメントの受け取りを望むことがある。動的ドキュメント (dynamic document) を生成する一つのメカニズムは、コモン、ゲートウェイ、インターフェース (Common Gateway Interface (CGI)) スクリプトを用いることである。クライアントは、CGIスクリプトにより、URL (Uniform Resource Locator) のフォームでドキュメントを要求する。この場合、URLはサーバ上のドキュメントでなくサーバのプログラムを参照する。サーバはそのプログラムによってドキュメントを生成し、ブラウザに該ドキュメントを返送する。サーバはその要求をドキュメントではなくてスクリプトを実行する要求であると同定し、かつ、サーバは、おそらくはURLの一部として認められた推論 (argument) を用いて、そのCGIスクリプトを実行する。これにより、ドキュメントを要求しかつ受け取るためのみに設定されたクライアントブラウザは、クライアントサーバ側のコードの実施の結果を表示することを許容することとなる。CGIスクリプトはブラウザをサーバ側のデータベースにインターフェース接続するのに用いられ、それによって、サーチの要求がURLとしてサーバに送られ、そのサーチ結果はドキュメントとして返送される。それらは、サーバ側で実行されかつブラウザは一切識別しないため、ユーザはCGIをいつ、どう使用するか制御することはできない。

【0006】もし、その計算機能 (computing) が、クライアント側のコード (アプレット) を書くために用いられるカリフォルニア (CA) マウンテン ビュー (Mountain View) のサン マイクロシステムズ社 (Sun Microsystems, Inc.) によって指定されたJ A V A言語 (Java Language) によるなどして、ドキュメントと共に伝送されるならば、ユーザはプログラムに実行についてより多くの制御手段を持つこととなる。

【0007】クライアント側でコードを実施することにより、クライアントはドキュメントを要求し、かつ、返送されたドキュメントはそのドキュメント中に埋め込まれたプログラムコードを含む。クライアントブラウザは、クライアント側のコードを知っていなければならないが、受領と同時にコードを実行する。クライアント側でコードを実行することにより、ドキュメントのアニメーション画像要素のような、そうしなければサーバ上で実施しかつサーバとクライアント間のリンク上に伝送するのに大きな帯域帯が必要となる、簡単なタスクを遂行するに最も適した使用方法であることが明らかとなる。

【0008】J A V AスクリプトとCGIとはその実行の場所 (locus) は異なるが、それ以外はそれほどでもない。サーバのオペレータはどのJ A V Aスクリプトがドキュメントに付されるか特定するのであるから、それらはなおサーバ制御されるプログラムである。これらの両方法は、いずれもサーバオペレータにより特定されるものである。それらは、ユーザに対する融通性を制

限している。サーバ側でのコードの実行は、サーバWeb側の開発者 (developer) によってセットされかつ制御されるので、読み取り側の自由裁量で行うことはできない。クライアント側でのコードの実行は、サーバが既にアプレットを含んでいるドキュメントに制限されかつアプレット認識ブラウザとともに用いる場合に制限されている。

【0009】ユーザに融通性及び制御を提供するための幾つかの試みがなされてきている。一つの試みは、専用のクライアントエージェント (Client Agent) を提供することである。クライアントエージェントは、蓄積された引用文献を収集し、録音された音響を演奏し、かつ、サーチを実施するようなことを行うため、ブラウザクライアントとして同じマシンで実施される。クライアントエージェントは、もし、ブラウザがクライアントを認識していれば、ブラウザクライアントと共に用いることができる。例えば、リアルオーディオ (Real Audio) のオーディオプレーヤーは、簡単なクライアントエージェントである。即ち、ネットスケープWebブラウザ (Netscape Web browsers) は、オーディオプレーヤー用フックを持っているので、このクライアントエージェントは、そのWebブラウザに遭遇したオーディオファイルにコンピュータスピーカへのルートを設定することができる。オーディオプレーヤーを認識できないWebブラウザについては、その結果は、ブラウザがオーディオファイルを受け取った時に、それをどう処理するのか理解できないということになる。クライアントエージェントは、また、クライアントとネットワークとを接続する帯域幅により、そののできるものが制限される。

【0010】ローミングエージェント (Roaming Agent) は、クライアント側又はサーバ側エージェントよりもより融通性があるが、広い利用をみていないようである。ローミングエージェントはエージェントの所有者によって特定されたタスクを行うプログラムであり、その実行位置は実行中にエージェントが見い出すものに応じて変わるものである。このローミングエージェントは、各ノードにおいて、ネットワークに十分な計算能力が存在する間は、特別の理由がない限り実行されない。インターネットの成長と商業化によって、安全性が問題であり、かつ、資源は容易に過負荷状態になる。その結果、極めて小数のサーバオペレータのみがシステム上でのローミングエージェントの実行を許可されている。近い将来には、サーバ上で動くことを許されるエージェントのみがサーバオペレータによって特に選別されることになる。そうしなければ、サーバはエージェント用の場所を求めるユーザの殺到によって急激に身動きできなくなるであろうし、又は競争相手のサーバ上でエージェントを動作させている企業でさえも、コンピュータの能力を“盗む”ことになろう。

【0011】種々のエージェントシステムの潜在的な間

題は、ユーザはサーバのオペレータの自由裁量で提供されたサーバ側のエージェント、又は、エージェントを認識できるクライアントととのみ対話しかつサーバオペレータによって管理されることが多いクライアント側のエージェントに制限されているので、平均的なユーザには一般化されたエージェントを実行する適当な場所はないということである。

【0012】サーバ又はクライアント以外の場所（locus）でユーザの情報を処理できる情報組織のための特別のシステムが提案されているが、これらのシステムは全てのサーバ及びクライアントが情報システムを認識していることを要するものである。WWWインダ（WWWind a）、プロスペロ アンド テーシーケイWWWロボット（Prospero and TckWWW Robot）がそのような情報システムの例である。

【0013】

【発明が解決しようとする課題】必要なものは、ワールド ワイド Webのようなドキュメント指向のフレームワークで好ましく動作し、使用されるサーバ又はクライアントとは無関係に、ユーザによって特定された位置で一般的なエージェントの実行を許容するコンピュータシステムである。

【0014】

【課題を解決するための手段】本発明によって改良されたドキュメント指向の計算システムが提供される。請求項1の発明は、クライアントからドキュメント要求を受け取り；サーバから応答を受け取り、トランザクション列中にトランザクションを生成し；トランザクション列中の各トランザクションを、トランザクションから情報を抽出し；トランザクションを変更し又はトランザクション列に付加すべき新規なトランザクションを生成する、一又はそれを越える整合エージェントに整合させ；トランザクションによって特定されるメソッド又は満足を与えるエージェントを用いて各トランザクションを満足させ、満足を与えるエージェント又はメソッドはクライアント及びサーバとは独立しており；整合エージェント、満足を与えるエージェント又はトランザクション列中のトランザクションによって指示されるように、サーバにドキュメント要求を送り；かつ、整合エージェント、満足を与えるエージェント又はトランザクション列中のトランザクションによって指示されるように、クライアントに応答を送る；各ステップからなるドキュメント検索及び操作方法である。

【0015】請求項2の発明は、請求項1の発明において、少なくとも整合のステップが特定のユーザのために保持された状態に応じて実施されることを特徴とする、特定のユーザのための状態を保持するステップからさらに成っている方法である。

【0016】請求項3の発明は、請求項1の発明において、少なくとも整合のステップが周辺機器のために保持

された状態に応じて実施されることを特徴とする、周辺機器のための状態を保持するステップからさらに成っている方法である。

【0017】請求項4の発明は、クライアントがドキュメント転送プロトコルによってサーバからドキュメントを要求するネットワークにおける、クライアント及びサーバと独立にネットワークの機能性を拡張する方法であって、特定のユーザのための状態を蓄積することを含む、特定のユーザによる使用のための個人エージェントをカスタマイズし；特定のユーザからのドキュメント要求を受け入れ、その際、ドキュメント要求は指定されたサーバに向けられかつ要求するクライアントから開始されるものであって；特定のユーザと協働する個人エージェントにドキュメント要求を送信し；指定されたサーバにドキュメント要求を送信し；ドキュメント要求を反映させるため状態を更新し；ドキュメント要求で要求されたドキュメントを、個人エージェントのところで受け取り；個人エージェントの状態によって指示されるようにドキュメントを変換する；各ステップからなる方法である。

【0018】請求項5の発明は、請求項4の発明において、特定のユーザは個人エージェントを用いる個人のグループの一人である、方法である。

【0019】請求項6の発明は、請求項4の発明において、ドキュメントを要求クライアントと異なるクライアントである受容クライアントへ送信するステップからなる、方法である。

【0020】請求項7の発明は、請求項4の発明において、特定のユーザによって要求されたドキュメントの履歴を、個人エージェントによって、蓄積するステップからなる、方法である。

【0021】請求項8の発明は、請求項4の発明において、エージェントが、転送のユニットがドキュメントであるハイパーテキスト転送プロトコルを用いるクライアント及びサーバと通信する、方法である。

【0022】請求項9の発明は、クライアント要求を周辺機器から提供するネットワークにおいて、ドキュメント転送プロトコルを用いてクライアントからドキュメント処理要求を送り；周辺エージェントのところでドキュメントを獲得し；周辺エージェントからの制御ドキュメントをクライアントに送信し、その際、制御ドキュメントはドキュメントの表示及びドキュメントに対して採る動作を特定する制御部を含み；周辺機器によるドキュメントの処理を開始するためクライアントからの制御命令を、ドキュメント転送プロトコルを用いて、クライアントから周辺エージェントに送るステップからなる、クライアントを周辺機器にインターフェースする方法である。

【0023】請求項10の発明は、多数のエージェントの表示を蓄積するエージェント配列；トランザクション

列；ドキュメント要求及びドキュメントをWebクライアント又はWebサーバから受け入れるためのプログラミング、及び、トランザクション列中に表示され、かつ、エージェント配列中の一又はそれを越えるエージェントによって動作されるトランザクションを生成するためのプログラミング含むハイパーテキスト転送プロトコルマネジャー；トランザクションをエージェントに整合させ、かつ、そのエージェントと整合するトランザクション列中のトランザクションを処理するために必要なとき、エージェント配列中のエージェントを活動させるためにエージェント配列及びトランザクション列に結合されるリゾルバ；からなる、クライアントとサーバの間で転送されるドキュメント及びドキュメント要求を変換するため、WebクライアントとWebサーバの間でネットワークに挿入されたWebエージェントである。

【0024】請求項11の発明は、請求項10の発明において、Webエージェントは周辺エージェントであり、周辺エージェントは、周辺機器によるドキュメント処理要求を仲介するため、周辺機器と一又はそれを越えるブラウザクライアント間に挿入され、周辺機器は周辺機器と対話するのに用いられる一又はそれを越えるブラウザクライアントから独立している、装置である。

【0025】請求項12の発明は、請求項10の発明において、Webエージェントがドキュメントファイリングシステムへのアクセスを提供しかつ制御する装置である。

【0026】請求項13の発明は、請求項10の発明において、Webエージェントは個人情報エージェントであり、個人情報エージェントは個人の状態を保持するため及び保持された個人の状態に応じてドキュメントとドキュメント要求を変更するため、ユーザのクライアントとネットワーク間に挿入されたエージェントである、装置である。

【0027】請求項14の発明は、請求項13の発明において、個人の状態が、個人情報エージェントのユーザによって動作される多数の独立したクライアントを包括する選択された資源の包括的個人ホットリストからなる、装置である。

【0028】請求項15の発明は、請求項13の発明において、個人の状態が、個人情報エージェントのユーザによって使用される選択された資源の包括的な個人履歴リストからなっており、包括的な個人履歴リストは個人情報エージェントのユーザによって動作される多数の独立したクライアントを包括するものである、装置である。

【0029】請求項16の発明は、ハイパーテキストの部分を実行可能なフォームにフォーマットし；リンクされた参照（link references）から機械読み取り可能なプリントシンボルを生成し；機械読み取り可能なプリントシンボルの表示をリンクされた参照用のアンカー

の位置に対応した位置でハイパーテキストドキュメントのプリント可能なフォームに挿入し；かつ、ハイパーテキストドキュメントのプリント可能なフォームをプリントする；各ステップからなる、ハイパーテキストドキュメントの一部を表すプリントされたドキュメント中にハイパーテキストリンクを保持する方法である。

【0030】請求項17の発明は、請求項16の発明において、機械読み取り可能なプリントシンボルがハイパーテキストドキュメントのプリント可能なフォーム中のアンカーの下にアンダーラインを形成する、方法である。

【0031】一実施例においては、使用される計算システムは、“エージェント”と指称され、ユーザエージェントのための実行位置を形成する。このエージェントは、一以上のWebと一以上のWebサーバとの間に挿入される。エージェントで実行されるエージェントはドキュメントを転送することによってクライアント及びサーバと対話する。エージェントはエージェントを認識できないクライアント又はサーバに対し透過でき、かつ、好ましくは、例えばクライアント又はサーバがエージェントを認識している場合であっても、サーバとクライアントの間で透過動作するものである。エージェントは、ドキュメント要求がクライアントからサーバに及びその逆に送られ、かつ、ドキュメントはネットワークを移動中に変換されることを必要とするか、又は、状態が保存されることを必要とする多くの応用例において利用することができる。

【0032】個人情報エージェントは、ユーザによって使用される種々のクライアントとこれらのユーザにアクセスされる種々のサーバとの間に挿入される。個人情報エージェントは、ユーザによってアクセスされた全てのドキュメントの履歴を保持する履歴エージェント

(History Agent)、ユーザが“ブックマークを付けた”ドキュメントのリストを保持するホットリスト (hotlist) エージェント、及び、ユーザクライアントに送付したドキュメントの複製を受領クライアントに送付するミラー (mirror) エージェントのようなユーザエージェント用の位置 (locus) を提供する。ユーザエージェントは、サーバからクライアントに向かう参照されたドキュメントにリンクを挿入することによってドキュメントを変換する機械読み取り可能なリンクエージェントのような変換エージェントであり、ここで、リンクはプリントされた後においてさえ機械読み取り可能なものである。

【0033】一般に、エージェントは、トランザクション上でエージェントを実行し、かつ、新規なエージェント及び新種のトランザクションの受け入れに対し解放である。エージェントによっては、トランザクション上で、特定のトランザクションを知覚しかつ変更するような比較的重要でない活動を行うものがあり、他方、トランザクションを遂行するか又はそれを転送するようなト

ランザクションを完全に操作するより主要な活動を行う別のエージェントもある。

【0034】ある実施例においては、Webエージェントは、Webクライアントを実行するようプログラムされたデジタルコンピュータとは別個の好適にプログラムされたデジタルコンピュータであり、他方、他の実施例においては、Webエージェント及びWebクライアントは単一のコンピュータで実行する二つのプロセスである。通常、帯域帯及び利用可能な計算能力で決定されるクライアント、エージェント及びサーバについての多くの特定の変更例は、この開示を読んだ後には、当業者に自明であるはずである。

【0035】Webエージェントの一つの特定の実施例は、ドキュメント指向周辺機器と対話するように設計されている。別の実施例は、複数のWebクライアント及びWebサーバについてユーザの状態及び履歴を記録する個人情報エージェントである。ユーザの状態を追跡することにより、種々のサーバから受け取った全てのドキュメントをユーザ提示用の一様なフォーマットに変換するための、個人エージェントによって使用される一組の好み (a set of preferences) とともに、クロスクライアント履歴リスト (cross-client history list) を作成することができる。

【0036】Webエージェントの一つの利点は情報表示と情報管理との相互関係を断ち、それによって、これらのタスクを独立のものにすることである。Webエージェントの別の利点は、ユーザに対しテキストよりもむしろドキュメントを扱うインターフェース又はグラフィック用ユーザインターフェースを許容することである。

【0037】Webエージェントのさらに別の利点は、HTTPのようなステートレスメッセージ通過プロトコル (a stateless message passing protocol) を用いた場合でも、ドキュメント転送の流れの中で計算動作を行うことができるということである。ドキュメントに埋め込まれたもの以外には、エージェント、クライアント及びサーバ間で交信すべき共有ステートがないため、実施は極めて簡単である。

【0038】本発明の本質及び利点の更なる理解は、明細書の残りの部分及び添付図面を参照することによって得ることができる。

【0039】

【発明の実施の形態】 ここで、本発明の実行するための好ましい実施例が、本発明の特定の使用例に関連して詳しく説明される。しかしながら、本発明は、それに限定されるものではない。当業者は、この開示を読んだ後、説明された実施例の他の有用な変更例を理解するであろう。

【0040】図1は、本発明が用いられる典型的なシステム環境におけるエージェント10のブロック図であ

る。ユーザはエージェント10に直接アクセスできるが、ユーザは典型的にはWebサーバ14にアクセスするためWebクライアント12である種の行動を起こすことによって、エージェント10にアクセスする。以下で説明するように、エージェント10は多くはWebクライアント12に透過性であり、同クライアント12はエージェントを見越してプログラムする必要はない。Webクライアント12及びWebサーバ14は、チャンネル16上でHTTPのようなハイパーテキストドキュメント透過プロトコルを用いて互いに通信するようプログラムされている。チャンネル16は、物理的チャンネルである。しかし、典型的には、論理チャンネルである。チャンネル16は、直接又はインターネットのようなネットワーク18を介してWebクライアント12とWebサーバ14を接続している。

【0041】Webクライアント12及びWebサーバ14の間には、エージェント10が挿入されている。ある場合には、ドキュメントは要求Webクライアント12でなく受領 (recipient) クライアント20にルートが定められている。受領クライアント20は能動Webクライアント12における活動の受動“コピー”を表示するクライアントとしてのような多くの利用方法がある。

【0042】エージェント10は、HTTPドキュメント要求とドキュメントを送受するので、それは、単にHTTPクライアント又はサーバと通信することを期待する、Webサーバ及びクライアントと対話することができる。図2は、どうやってこれを行うかを示すエージェント10のより詳細な図である。図2は、主にHTTPを介してクライアント及びサーバと対話するエージェントを示すが、エージェントは他のプロトコル及びWebのない環境において使用するように適用できることを理解されたい。

【0043】エージェント10は、外部ソースから受領エージェント30及びHTTPインターフェース28を経てクライアント及びサーバからトランザクション (17, 17a, 17b等) を受け入れるものとして示されている。エージェント10はエージェント30を用いてトランザクションを処理するリゾルバ (resolver) 24を含んでいる。リゾルバ24は、エージェント配列 (array) 20、特徴計算機 (feature calculator) 21、トランザクション列 (queue) 23、マッチチェッカー (match checker) 25、“act_on”プロセッサ27、ハンドラ (handler) 29及びエージェントレジストラル (registrar) 31を含んでいる。

【0044】エージェント10は、インストールすべきエージェントを備えたトランザクションを制御する“エージェントインストール”エージェントのような既存の標準的な組をなすエージェントで例示可能であるが、エージェントはトランザクション上で動作可能で

ある前に、エージェント配列 20 中に 1 又はそれを越えるエージェント 30 を有していなければならない。エージェントレジストラル 31 (それ自身エージェントインストールエージェントである) は、エージェント 30 を受け入れ、かつ、それらをエージェント配列 20 上に置く。図 2 に示すように、各エージェントは、少なくとも一つの基準 (criterion)、act_on() メソッド、及び、handle() メソッドを有しており、その使用方法については、以下で説明する。エージェントインストールプロセスの一部として、エージェントはリゾルバ 24 によってその基準を“登録 (register) する”。それからリゾルバ 24 は、基準によって索引が付されたマッチテーブル (match table) 中に登録された基準を蓄積する。これによって、任意の所定の基準又は特徴に対して関与するエージェントのリストを迅速かつ容易に検索することを可能にしている。

【0045】リゾルバ 24 の役割は、トランザクションを受け入れかつエージェントにトランザクションと何らかの作用を行うようにさせることである。エージェントリゾルバ又は“ハンドル (handles)”によって動作が起こされると、トランザクションはリゾルバ 24 から削除される。そのようなエージェントの動作は、handle() メソッドと指称され、トランザクション上で動作するが、しかし、トランザクションを完全には制御しない act_on メソッドとは区別される。エージェンシーを開放されかつ拡張自在にするため、操作可能なエージェント、*

テーブル 1 トランザクションの要素

要素	説明
Name/ID	エージェンシー中のトランザクションを独特に同定する；プロセス ID に類似している。
From_Machine	トランザクションのソース；クライアント、サーバ又のエージェンシーである得る；もし適用しなければ、この要素は例示されないか、又は、無効値で例示される。
To_Machine	トランザクションの行き先；クライアント、サーバ又のエージェンシー；もし、適用しなければ、この要素は例示されないか、又は、無効値で例示される。
Handler_Queue	このトランザクションを制御できるエージェントへのポインターのリスト
Default_Handler()	このトランザクションを率先して制御するか又は制御可能なエージェントがない場合、それはそれ自身を制御するための命令 (instruction) (方法) を備える。
Feature_List	このトランザクションに適用可能な特徴値のリスト
Data	トランザクションのデータ

【0048】トランザクションがリゾルバによって受け取られると、それは最初に特徴計算機 (feature calculator) 21 によって処理される。特徴計算機 21 は、特徴セットを見つけるために、トランザクションのデータ要素 (及びおそらく他の要素も) を走査することによって、トランザクションに対する特徴リストを生成す

*それらの方法又はトランザクションは所定の時間より前にはセットされない。この理由から、リゾルバ 24 はトランザクションを処理するに際し、機敏でなければならない。リゾルバ 24 はどのトランザクションがどのエージェント上で動作されるか又は満足されるか特定しない。その代わり、エージェント配列 20 中のエージェントはどのタイプのトランザクション上で動作又は処理するか特定し、かつ、トランザクションはそれら自身の特徴を特定する。このようにして、リゾルバ 24 はトランザクションを、エージェントに含まれる基準及びトランザクション中に含まれる特徴に基礎を置くエージェントに整合させる。このコンセプトは、エージェンシー 10 の動作の説明に関連して、以下で説明する。

【0046】ここで、エージェンシー 10 の動作を説明する。例示に際して、エージェンシー 10 は、既にそのエージェント配列中に、エージェントインストールエージェントのような少なからずの標準エージェント 30 を持っている。クライアント及びサーバがエージェンシー 10 にトランドクシオン 17 を送ると、このトランザクションは HTTP インターフェース 28 によって受け入れられる。典型的なトランザクションは、テーブル 1 に示されている。しかしながら、全ての要素が全てのトランザクションに存在するわけではないことを理解されたい。

【0047】

る。トランザクションの特徴の例は、テーブル 2 に示されている。特徴リストは、トランザクションの特徴の“場所 (cache)”である。全ての特徴を一度評価することによって、トランザクションがある特徴を持っているか否かについてエージェントが知る必要がある度ごとに、トランザクションデータを走査しなければならない

ということはない。もし、トランザクションが変更されれば、勿論、トランザクションは特徴計算機21又は同等の手順によって再処理されることになる。

【0049】トランザクションの各特徴は、特徴計算機21において、コードの断片(C又はPerl関数等)によって表され、その結果、付加的な特徴は、特徴計算機の既知の特徴セットにいつでも追加することができる。例えば、もし、エージェントが所定の位置に配置された後、新規なグラフィックドキュメントフォーマットXYZが出てくると、新規な特徴断片IS_XYZを特徴計算機21への挿入のために(おそらくは、“特徴インス

トーラ”エージェントで指示されるトランザクションを*

テーブル2 トランザクション特徴

特徴	説明
is_response*	このトランザクションは要求に対する応答である。
is_request*	このトランザクションはドキュメント用要求である。
is_agent_response*	このトランザクションはエージェント(エージェントはトランザクションを生成できる)からの応答である。
is_proxy_request*	このトランザクションはプロキシ(proxy)から/への要求である。
is_agent_request*	このトランザクションはエージェントから/への要求である。
is_text	このトランザクションのデータはテキストドキュメントである。
is_html	このトランザクションのデータはHTMLフォーマットドキュメントである。
is_image	このトランザクションのデータはイメージである。
is_local_source	このトランザクションはエージェント内のソースからのものである。
client_is_netscape	このトランザクションを扱うクライアントはNetscapeブラウザ又はそれと両立するブラウザである。
is_file_request	このトランザクションはファイルに対する要求である。
is_interform	このトランザクションはインターフォーム(一つのフォームでプログラムと結合するドキュメント)である。

*これらの特徴は、各々のトランザクションに対するデフォルトによって存在する。

【0051】特徴リストが一度生成されかつトランザクションと関連付けられると、トランザクションは、トランザクション列23に蓄積される。トランザクションはそこからマッチチェッカー25によって検査される。オブジェクト指向環境においては、マッチチェッカー25の機能性は各々のエージェントの部分である一組のマッチチェック用メソッドとして実施される。いずれにしても、各々のエージェントはどのトランザクションに整合(マッチング)するのかを見るために、各々トランザクションに整合される。整合はエージェントに特有のブール代数表現であって、ここで、その表現は特徴の関数である。例えば、エージェントに対する基準は、
is_file_request=true&&is_local_source==false
となろう。

*用いて) リゾルバに送ることができる。それから、トランザクションのデータを走査するとき、特徴計算機21は、もし、XYZフォーマットを検知すると、新規なコードの断片は、“true”のリターン値(return value)を与え、かつ、特徴計算機は該トランザクションに対する特徴リストにIS_XYZを付加する。以下で説明するように、エージェントは各々それら自身の基準(criteria)を備えているから、XYZフォーマットドキュメントに作用し、かつ、処理するようにプログラムされたエージェントはエージェント配列に容易にインストールできる。

【0050】

【0052】エージェントに関与(interest)するこれらのトランザクションは、エージェントに対する基準表現(criteria expression)で示される。もし、基準表現が“false”であれば、エージェントはトランザクションに関与しない。もし、基準表現が、“true”であれば、そのときは、エージェントはトランザクションに関与する。このように、上記の基準表現に対し、もし、トランザクションの特徴リストが、トランザクションがファイル要求であり、かつ、それが非ローカルソース(nonlocal source)であることを示しているとする、その場合には、エージェントはトランザクションを処理する。ある実施例においては、エージェントはact_on()メソッド及びhandle()メソッドに対し異なった基準表現を持つことができる。効率的な実施においては、各々のエ

エージェントは各々のトランザクションを考慮しない。代わりに、各エージェントは特徴及びこれらの特徴を有するトランザクションへの参照をソートしたテーブルである特徴インデックス33を考慮する。その方法により、各エージェントには関与するトランザクションを走査することのみが必要となる。勿論、トランザクションを処理する際に、エージェント特徴インデックスも用いて、無関係なエージェントを無視できるようにすることができる。

【0053】もし、エージェントが（全体の基準表現又は特定のact_on基準表現のいずれかによって示された）トランザクションにアクトオン（act on）することに関与すると、エージェントのact_on()メソッドが実施される。これは、act_onプロセッサ27によって行われるか又はエージェントのact_on()メソッドの実施によって論理的に行われる。スレッドドプロセス（threaded process）が用いられる場合には、各々のact_on()メソッドは並行処理を可能にするための分離したフォークドプロセス（forked process）である得る。

【0054】act_on()メソッドは、I/Oアクションを起こすこと、計算を行うこと、トランザクションを付加するか又はトランザクションのデータ、特徴セット、又は、他のデータ要素を変更することを含む。好ましい実施例においては、I/Oアクションはhandle()メソッドのために保留されており、それによってすべてのact_on()メソッドは、I/Oコール（call）のための処理妨害を心配することなく（同期）コールをブロックし続けることができる。もし、トランザクションがI/Oに関係するが、そのトランザクションを完全には処理（handling）しないある処理を必要とするならば、その場合には、元のトランザクションの非処理I/O（the nonhandling I/O）が新規なトランザクションの処理によって生成される。

【0055】トランザクションがすべての関与エージェントによって一度アクトオンされると、トランザクションは（実際又は論理的のいずれかにより）ハンドラ29に進む。act_onプロセッサ27とともに、ハンドラ29は論理構造であり得、制御は単にエージェントのhandle()メソッドをコールすることによって行われる。一実施において、トランザクションを処理することに関与する各エージェントは、そのhandle()メソッドのポインターをトランザクションのハンドラの列（queue）にフックする。効率化のためにエージェントのact_on()メソッドは、act_on()メソッドがコールされたときフックを付加することができる。エージェントは、それ自身のhandle()メソッド以外にフックを付加することができる。例えば、エージェントは、コード断片（code snippet）上で別のトランザクションにフックを付加することができる。

【0056】全ての関与エージェントがハンドラ列中に

フックされているときは、ハンドラ29はその列を通して歩進しかつ順にhandle()メソッドを実施する。もし、エージェントがそれがトランザクションを実際に完全に処理できることを見出した場合、そのhandle()メソッドは、“true”リターン値（return value）を返し、かつ、ハンドラ29はそれにふさわしく列（queue）23又はハンドラ29からトランザクションを削除する。もし、エージェントがそれがトランザクションを完全には満足させないことを見出したとすれば、エージェントのhandle()メソッドは、エージェントが非制御トランザクションに対して採るべき任意のステップを実行し、かつ、“false”のリターン値を返し、それによってハンドラ29が操作のための次のエージェントについてチェックを進めるようにする。いずれのエージェントも“true”を戻すことなくトランザクションのハンドラ列の最後まで達すると、ハンドラ29はトランザクションのデフォルトhandle()メソッドを実施する。理解されであろうように、エージェンシー10及びリゾルバ24のこれらの各々の活動的な要素は、切り離した計算装置又は単に一般目的の計算装置によって実施される異なったプロセス又はルーチンのいずれかであり得る。

【0057】トランザクションが一以上の宛先（destination）に向う場合には、一つのエージェントのact_on()メソッドはそのトランザクションのクローン（clone）をつくり、かつ、マシンフィールドへ向かう各トランザクションを一つの宛先に向けてセットすることができる。

【0058】以上で説明したように、トランザクション17は、リゾルバ24によって受け入れられ、トランザクションの特徴は、該トランザクションと関連する特徴リストに貯えられ、かつ、トランザクション17は、トランザクション列上の所定の位置に配置される。この特徴リストは、特徴インデックス33と相互参照される。それから、リゾルバ24は、基準表現を有するエージェント配列20中で、トランザクション17の特徴で評価したとき、“true”と評価されるエージェント30を同定する。これらのエージェント30は、トランザクション17上で作用して、I/Oを実施し、計算を行い、蓄積状態又は内部データを変更し、新規なトランザクションを生成し、トランザクション17を変更する。リゾルバ24は、act_on()メソッドの部分として又はそれとは別に、トランザクションのためのハンドラの列を構築し、かつ、handle()メソッドが“true”の値を返す、又は、もし、handle()メソッドが“true”の値を返さないならば、デフォルトハンドラを実施するまで、列中の各々のハンドラを順に実施する。このように、エージェンシー10がセットアップされた後に、処理を行っているエージェントが生成される場合であっても、あるいは、エージェンシー10がセットアップされた後に新規な特徴が実施される場合であっても、トランザクションはエ

ージェンシー 10 によって処理可能である。

【0059】 エージェントは動作の巨大な配列 (vast array of operation) を実施でき、かつ、他のエージェントと協働して独特な性格を持ったージェンシーを形成するよう構築することができる。エージェントの例としては、サーチ結果にフィルターをかけるための、検索されたドキュメントから望まないものを取り除くための、ドキュメントにボタン (buttons) を付加するための、履歴情報を蓄積するための、ユーザの状態を蓄積するための、又は、ユーザの環境設定 (configuration) 及び初期設定 (preference) を蓄積するためのエージェントがある。

【0060】 実施されるように、ージェンシー 10 は、新規なエージェントがエージェントインターフェースの標準に一致している限り、ージェンシー 10 が構築されたときには、予期していなかったホストエージェントであり得る。これらのインターフェース標準 (standard) は、オブジェクト指向プログラミングの使用により実施され、その際、“エージェント”のラベルを付したオブジェクトのクラスを特定することができる。エージェントは少なくともトランザクションを変更し又はトランザクション列 22 に付加するための機能を含んでいる。

【0061】 HTTP インターフェース 28 を介して受け取る要求及びデマンドは、直接又は HTTP インターフェースエージェントを通すかのいずれかにより、トランザクションに変換される。各トランザクションは、トランザクションのソース、トランザクションの宛先及びトランザクションに関連するデータを同定するフィールドを含んでいる。ージェンシー 10 は、ドキュメント指向計算システムで動作するので、データはドキュメント又はドキュメントの参照のフォーム (形式) である。パッキング要求 (packing request)、応答及びトランザクションに加え、HTTP インターフェース 28 は、いかにして特定のマシンに到達できるかを記述し、かつ、マシンがどのブラウザを使用しているかを記述したマシンのデータベース 28 の情報に基づいて応答 (レスポンス) 及び要求を変換することもできる (即ち、そのようにして HTTP インターフェース 28 は、適当なブラウザに一致させるため応答 (レスポンス) を変更できる)。

【0062】 リゾルバ 24 の処々の要素 (エレメント) は、実施形態により、可視要素又はオブジェクト参照 (object reference) である。例えば、オブジェクト指向プログラミング環境においては、act_on プロセッサ 27 の機能は、トランザクションに作用する各エージェントにおけるメソッドとして内蔵されている。これと同じことがマッチチェッカー 25 及びハンドラ 29 でも真実である。さらに、トランザクションは、実際には、場所から場所に移動するのではなく、トランザクシ

ョン列 23 上で動作できるものである。実際の要素を移動させ代わりに、ポインタをデータ要素へ向ける操作、メソッド又は機能で十分である場合には、ポインタは実際のデータの代わりに図 2 の回りで動かされるデータである。

【0063】 種々のージェンシーをここで説明することとする。一つの包括的なージェンシーは、人が欲する任意のージェンシーを動かしかつそれらの対話 (interaction) を制御するまさに個人的なージェンシーである。

【0064】 図 1、図 2 に示すシステムは、バイト (byte) 一又はファイル指向ネットワークで使うことができる。しかし、その利点の多くは、ドキュメント指向ネットワークで用いることから引き出されるものである。バイト指向ネットワークにおいては、全てのオブジェクトはバイトセットによって規定され、従って、オブジェクトは一般にそれらが含むバイト又はそれらが入力又は出力するバイトによって見ることができる。ファイル指向ネットワークにおいては、オブジェクトはファイルとみなされる。ファイル指向ネットワークの一般的な例は、ファイルサーバが設置されたローカルエリアネットワーク (LAN) である。LAN は遠隔のファイルをローカルファイルのように見せるため、ファイル操作のローカルな動作システムのコマンド (リストディレクトリ、チェンジディレクトリ、ムーブ/コピー/ディレートファイル等) を書き換える (エミレート (emulate) する)。Unix のようなある種のファイル指向動作システムにおいては、ファイル操作コマンドを全く簡単に周辺機器の操作に適用できるよう、周辺機器は、ファイルに見えるように作られている。従って、プリンタでのフィルのプリントは、ファイルをプリントを表すファイルに複写することによって行われる。

【0065】 ドキュメント指向ネットワークにおいては、全てのオブジェクトがドキュメントのように見える。リンクで結合されたコンポーネント (構成成分) を備えた複合ドキュメントについては、全ての Web サーバ上の全てのドキュメントは、ユーザに対して、一つの大きなリンクされたドキュメントとして現われる。Web エージェンシーは、このコンセプトをさらに進めたものであって、ネットワーク上でアクセス可能な殆ど全てのものが一つのドキュメントとして出現するようになっている。一度ドキュメントが計算用ユニット (the unit of computing) であるとなると、ドキュメントをより容易に紙等価物に書き換える (エミュレートする) ことができ、その際、ドキュメントは以前に紙を扱った人なら誰にも慣れた範例 (paradigm) を使って、余白に書き込まれ、蓄えられ、郵送され、保管所に保管される等のことを行うことができる。ドキュメントを中心とする一つの副次的効果は、ドキュメントを扱うツールが相互交換できるようになるということである。紙ドキュメント

領域における類似性 (analogy) は、ペンと机が相互交換可能な用品となり、ドキュメントは任意のペンを用いて編集されかつ任意の机の上に貯留しておくことができるということである。

【0066】ドキュメント指向計算システムは、ドキュメントを自分自身の中を通過させるエージェンシーのネットワークのようにみえる (図1, 参照)。このシステムにおいて、それらが全て適当な複合ドキュメントであるため、入、出力及び蓄積装置は簡単に扱えるようになる。入力装置は、ドキュメントソースであり、そのため、それらは入力装置からドキュメントを要求するエージェンシーから眺めたときには、常に更新され、永久に拡張し続けるドキュメントの集合である単一のドキュメント、あるいは、一又はそれを越える宛先へドキュメントを常に送り続けるクライアントとして現われる。出力装置は更新でき、常に拡張するドキュメントとの集合である一つのドキュメント、又は、一つ又はそれを越えるソースから常にドキュメントを要求するクライアントとして現われるドキュメントシンク (sinks) である。蓄積装置は、ドキュメントを蓄積装置に送りかつその後検索される入力装置及び出力装置の両方の様相を有している。

【0067】図3は、簡単なエージェンシーの論理ブロックダイアグラムである。エージェンシー32は、クライアントA、クライアントB、サーバA、サーバB及びサーバC間に挿入されている。図示するサンプル要求は、クライアントBがクライアントAが何を見ているかを“フォローしながら”、又は、モニタしながら行うドキュメント用の簡単な要求である。クライアントAは、エージェンシー32にドキュメント要求を送る (線34で示す)。ドキュメント要求は、プロキシエージェンシー36によって処理され、それがドキュメント要求をサーバAに送り (線38)、サーバは要求されたドキュメントをプロキシエージェンシー36に戻す (線40)。重要なことは、クライアントAは、プロキシエージェンシー36がそれとサーバAの間に挿入されたことを知る必要がなく、また、サーバAはプロキシエージェンシー36を知る必要がないということである。このことが既存のクライアント及びサーバをエージェンシー32とともに利用可能にしている。線34、38及び40は、HTTPチャンネルを規定している。

【0068】クライアントAからのドキュメント要求は、実際には、最初に、ドキュメント要求を備えたトランザクションをトランザクション列上に押し上げるエージェンシー32のHTTPインターフェース (図示せず、図2, 参照) によって受け入れられる。プロキシエージェンシー36は、ドキュメントを直接要求するか、又は、ドキュメントを検索させるトランザクションを生成することによるかのいずれかにより、ドキュメント要求を処理するエージェンシーである。いずれによっても、

検索されたドキュメントは、ドキュメントが受け取られたときには、トランザクションとして列 (queue) 上に押し上げられる。該トランザクションは、リゾルバによってホットリストエージェンシー42及びフォローエージェンシー44と整合されるが、しかし、トランザクションをクライアントに送るデフォルトトランザクションによって処理される。このようにして、ホットリストエージェンシー42とフォローエージェンシー44は、単にトランザクションを“アクトオン (act on) し”他方プロキシエージェンシー36は、トランザクションを処理する。勿論、フォローエージェンシー44は、I/Oを実行することを要求されるので、リゾルバは単にフォローエージェンシー44をオリジナルトランザクションに“アクトオン”させるというのではなく、フォローエージェンシー44によって処理される新規なトランザクションを生成することができる。

【0069】ホットリストエージェンシー42は、クライアントAに受け渡されたドキュメントを検査し、かつ、エージェンシー32によって保守されるホットリストデータベース中にドキュメントへの参照を蓄積する。このホットリストデータベースは、ドキュメント要求を送るクライアントによって、後に一つのドキュメントとして、再びアクセスされ、リゾルバはそのドキュメント要求をトランザクションのハンドラとしてホットリストエージェンシー42に整合させる。

【0070】フォローエージェンシー44は、クライアントAに受け渡されたドキュメントを検査し、かつ、クライアントBについてトランザクションを生成することにより、コピーをクライアントB宛先にして送る。これにより、クライアントBのところのユーザ又はコンピュータは、クライアントAのところの何が起きているかをフォローすることができる。フォローエージェンシー44の一つの応用は、他のユーザを訓練することである。別の応用は、クライアントによって検索される全てのドキュメントのローカルな蓄積である。これは、クライアントAが検索された全てのドキュメントを蓄積する能力を構築されているか否かに拘りなく行うことができるということを銘記されたい。図3に示す例において、クライアントBは、図1に示す受領クライアント20のような“受領 (recipient) クライアント”である。

【0071】ホットリストエージェンシー42と同様に、フォローエージェンシー44は、エージェンシー32に通常のHTTPドキュメント要求を送ることにより制御することができる。例えば、エージェンシー44によって供給される各ドキュメントは、“Stop Following”, “Rewind”, “Refresh”, “Alert”, “Option”等のラベルが付されたHTTPボタンを含んでいる。従って、各ページは、付加ボタンをそのページ自身に付した状態でクライアントBの前に現われる。ボタンに対しURLを適当に割り当てることにより、押された時、それらは適

当なメッセージをフォローエージェント44に送る。

“Stop Following” ボタンをクリックすることにより、クライアントBは、HTTPにドキュメント要求を送り、それがURLを参照し、かつ、それに対し、リゾルバはフォローエージェント44によって処理可能なトランザクションを生じる。トランザクション列22上のトランザクションが処理されると、フォローエージェント44は、トランザクションを受け取り、それに従って反応する。“Rewind” ボタンに対しては、フォローエージェント44は、ローカルな置き場所 (cache) からドキュメントを呼び戻しかつ再送付する。“Alert” ボタンに対しては、フォローエージェント44は警報 (alert) の特徴を表示するクライアントAにドキュメントを送る。“Option” に対しては、フォローエージェント44は、クライアントからの更なる入力を得るために、通常、HTTPフォーム (形式) のドキュメントをクライアントBへ戻す。

【0072】次に、図4を参照すれば、エージェントとのより複雑な対話セットが示されている。すでに説明したプロキシエージェント36、ホットリストエージェント42及びフォローエージェント44に加えて、図4は、また、全てが通常のHTTPチャンネルであり得る多数のエージェンシー32に対する外部接続部46を示している。インタレスト (interest) エージェント50、フィルタエージェント52、スケジュールエージェント54、及び、インストールエージェント56もまた図示されている。インタレストエージェント50は、ユーザの関心を示す活動を遮って、かつ、これらの関心に関連する追加ドキュメントを自動的に集める。フィルタエージェント52は、プロキシエージェント36によって検索されたドキュメントを介してシフト (桁移動情報, shift) を受け取り、ドキュメントをクライアントAに戻す前にフィルタールールに従ってそれらを変更する。フィルタエージェントは、望まない細部を取り除き、または、嫌な資料 (material) を除くのに用いることができる。スケジュールエージェント54は、ドキュメントを受け取りかつそれらをマルチスレッド (複数起動, multithreads) し、それによって、ユーザがドキュメントをロードするのに待たないようにしている。もし、サーバとエージェンシー間の接続がエージェンシーとクライアント間の接続よりも遅いときには、スケジュールエージェント54は (おそらく、インタレストエージェント50の援助によって)、ユーザがフォローしようとするリンク及びエージェンシーにおけるそれらドキュメントの貯留場所を予知しようとする。インストールエージェント56は、特定のエージェントをエージェント配列中にインストールするための要求を受け取る。明らかなように、クライアント、サーバ又はエージェンシーそれ自身の設計又は実施においてさえ意図しなかったタスクであっても、それら多くのタスクを処理するた

めの他の多くのエージェントをエージェンシー32によって例示することができる。

【0073】図4に示すようなエージェンシー32を用いたドキュメント検索動作を次に詳細に説明することとする。クライアントAは、HTTPドキュメント要求 (通常、URLをその要求の一部として含んでいる) によって、ウインドウAからサーバAからのドキュメントを要求する。要求は直接サーバAに送る代わりに、エージェンシー32に送られる。エージェンシーHTTPインターフェース (図示せず) は、要求をエージェンシーのトランザクション列 (これまた図示せず、図2、参照) の所定の位置に配置されたトランザクションにフォーマットする。エージェンシーのリゾルバは、ドキュメント要求トランザクションをプロキシエージェント36に整合させ、かつ、ドキュメント要求を送出しかつサーバAからドキュメントを検索する。ドキュメントを受け取ると、それに対してトランザクションが生成され、かつ、トランザクション列上に配置される。図3で示したように、リゾルバはトランザクションをホットリスト42及びフォローエージェント44に整合させる。リゾルバは、また、トランザクションをインタレストエージェント50に整合させ、かつ、トランザクションを処理するフィルタエージェントを有している。インタレストエージェント50は、ユーザがどのタイプのドキュメントを検索しているのか認識するためトランザクションを走査し、かつ、それからドキュメント要求をトランザクション列上に押し上げることによって、それらのドキュメントを独立に獲得する。関心のあるドキュメントは戻され、かつ、インタレストエージェント50のところで指示されたトランザクションにフォーマットされる。インタレストエージェント50は、トランザクションをスケジュールエージェント54に渡すことによって、トランザクションを処理し、スケジュールエージェント54は、ウインドウBでクライアントAに関心のある項目リストを表示する。トランザクションエージェント整合を適切に整えることにより、スケジュールエージェント54は、フィルタエージェント52からウインドウAに戻る各ドキュメントに変わることができ、かつ、その情報を使って、ドキュメントがクライアントによって検索されたとき、関心項目のリストからその参照を削除するようスケジュールテーブルを更新することができる。

【0074】インストールエージェント56は、考慮すべきエージェントのリゾルバリストへのインストール又は収容のための、エージェントに対する要求であるトランザクションに整合している。

【0075】Webエージェントの構造及び動作について説明したので、次に、Webエージェントの多数の例証について説明する。そのエージェントの一つは、図4で示したのと類似した個人情報エージェントである。個人情報エージェントは、ユーザとネットワークの対話全

てにわたり包括的に動作するドキュメントのモニタ及び情報フィルター (intelligent filter) として働く。例えば、ユーザは単一の包括的の履歴リストを持ったままで、種々の異なったブラウザを種々の異なった場所で使いたいと思うことがある。これは、個人情報エージェントによって実現される。個人情報エージェントは、ネットワーク上のどこにあっててもよく、好ましくは、ユーザがWeb上でドキュメントを見たいと思う全ての位置から到達できる場所にある。Webエージェントへの入出力は、通常、HTTPであり、ユーザは任意のブラウザを用いてかつ有利な結果を得ることができる。もし、ユーザがWebのページデザイナーであれば、フォローエージェントを備えたWebエージェントは、ユーザが同時に多数の異なったブラウザを用いてドキュメントを見ることを可能とし、それによって、ユーザが種々のブラウザにおいて、一つのドキュメントがどのように見えるのかの認識を得ることを可能にする。

【0076】個人情報エージェントは、個人の注釈

(これは、エージェントがドキュメントを検索する毎に、ドキュメントに付加される) をドキュメントに付加することをユーザに許容する、ユーザのための外部ドキュメントを特別に作るのにも用いることができ、又は参照のための全ての到来ドキュメントを走査することによって、ドキュメントを貯留し、かつ、ユーザの要求がある前に参照されたドキュメントを得るのに用いることができる。エージェントの追加機能は、また、ドキュメントを検索する際のドキュメント上のボタンとして付加することができる。

【0077】Webエージェントの別の応用例は、周辺機器、プリンタ62及びドキュメント中心ネットワーク間に挿入された図5に示すプリンタエージェント60のような、周辺エージェントである。プリンタエージェント60は、ブラウザ64上でプレビュー頁70の部分として、プリントプレビューイメージを表示させるとともに、ユーザから要求されたときに、ドキュメント66をプリントするのに用いることができる。入力、出力及び蓄積装置により、これらの装置の制御及び環境設定は、ドキュメントを介して、おそらくは、テーブルとフォーム(形式)を含むドキュメントを介して行われる。このようにして、周辺機器は、Webサーバを書き換える(emulate)ことができ、ドキュメントの蓄積又は出力に対するHTTP PUT要求及び環境設定フォームのためのHTTP POST要求のための宛先になり得る。周辺機器は、また、(出力すべきドキュメント用の) HTTP GET要求及びPUT又は(ドキュメント入力用の) POST要求用のソースである、Webクライアントをも書き換える(emulate)ことができる。周辺機器はまたプロキシとしても使える。

【0078】周辺機器と異なり、周辺エージェントは、ドキュメント又はドキュメントに対する要求を変換

する(即ち、応答を再発送するための一方のクライアントからの要求を他方からの要求に変換する)ことができる。このようにして、エージェントは、演算資源とともに、入力装置、出力装置、蓄積装置を管理するため、周辺機器(多くのユーザ)の代わりに活動するユーザ周辺エージェントの代わりに動作するより複雑でかつ新規なWebクライアントを表している。再び、図5を参照すると、プリンタエージェント60は、プリンタ62へのドキュメント中心インターフェースを提供するに用いられている。ユーザがブラウザ64又はその他を介して、HTMLドキュメント66をプリンタ62でプリントすることを要求するとき、ユーザは、プリンタエージェント60に単にドキュメント66に対する要求を送り、プリンタエージェント60はドキュメントを獲得し、かつ、それをプリンタ62上に出現するようにし(即ち、レンダリングの低解像度バージョン)、制御ボタン68をドキュメントに取り付け、かつ、ドキュメントのプリントプレビューバージョンをユーザに提供する。図5に示すように、ブラウザ64の出力70は、

“PRINT”, “OPTION” 及び “SATATUS” のラベルの付いた制御ボタン68を含んでいる。“PRINT” ボタンは、プリンタエージェント60によりドキュメント62をプリンタ62に送るタグを含んでおり、他方、“OPTION” 及び “SATATUS” ボタンは、プリンタエージェント60によって、それぞれオプション選択フォーム及びプリンタステータス頁を提供させるものである。

【0079】Webエージェントは、標準Webプロトコルを用いて通信するので、全てのエージェント、クライアント及びサーバは共通のプログラム言語、インターフェース又は動作システムを共有している必要はない。このようにして、プリンタエージェント60とブラウザ64は、それらが異なったプラットフォームにあるとしても対話することができる。実際、ブラウザ64は、エージェントのコンセプトを知らないとしても、プリンタエージェント60と対話できる。プリンタ用の全てのドキュメントは、プリンタエージェント60に送られるドキュメント要求によって、ユーザに提供できるので、周辺エージェントは、ドキュメント要求を処理するのに理想的でもある。プリンタエージェント60は、プリンタのロケーションを示すWebページをユーザに表示することによって、ユーザにプリンタを選択させるような付加的な環境設定タスクを処理することができるとともに、プリンタのクリック自在なマップを含んでいる。

【0080】図5に示すように、プリンタエージェント60は、制御装置によって原ドキュメントにはないプリントプレビューページをフォーマットする、すなわち、前述したボタン制御部68及びページのURLを表示する制御部を事前に見ることができる。このことは、ドキュメントをプリントしかつプリントを事前に見るた

め、ユーザがブラウザの一部として設けられたボタンを使用する必要があった従前のブラウザに対する改良である。ブラウザの全てが全てのプリンタを扱うようプログラムできる訳ではないので（現在のところ、真のプレビュー機能を有するブラウザではない）、ユーザはプリントするページそのものを見ることはできないが、ブラウザが仮定したとおりにプリントすることができる。レンダリングを行うプリンタエージェンシーにより、より正確なプリントプレビューが提供され、他方、ブラウザ製造者は多くの異なったプリンタに対してプリンタエミュレータ（printors emulators）をコード化しなければならないことから解放される。プリンタは機能上、ブラウザ又はホストソフトウェアへ変更を加えることなく変更することができる。

【0081】プリンタエージェンシーにより、HTMLフォーム（又は、該事項に対する任意の他のフォーム）のドキュメントは、プリンタへ送られ、かつ、Postscript及びAcrobat言語のようなページ記述言語の場合と同様に、ドキュメントの著者の代わりにユーザによって提供されるセッティング（調整）によって再生される。ドキュメントをプリンタエージェンシーに移送するのに必要な帯域幅も、それがHTMLフォーマットであるため減少している。ドキュメントはプリンタエージェンシーに到達したときには、まだ、HTMLフォーマットであるので、プリンタエージェンシーはドキュメントに対しある程度巧妙な処理を行うことができる。例えば、ユーザはプリントすべきドキュメントには一以上のレベルのリンクされたドキュメントが随伴すべきことを指示することができる。無限のリンクセットを持つドキュメントに対しては、ユーザはプリントエージェンシーが全てのドキュメントの“巻（book）”をプリントすることを指示することができる。プリンタエージェンシーは、ユーザのクライアントに対し透過性であるので、それは、レンダリング及びプリントのために全ての必要なドキュメントを透過的に検索することができる。プリンティングエージェンシーは、また、内容のテーブルのようなドキュメント流（ストリーム（stream））の部分として新規なドキュメントを作ることができる。

【0082】図6は、プリンタエージェンシーを用いて何を行うことができるかの一例を示している。プリントすべきドキュメントは、Postscriptファイル又はバイナリーファイルのようなレンダリング命令に対するのとは反対に、プリンタエージェンシー60は、ドキュメントを解釈できない。図6に示すように、ドキュメントは翻訳されかつページの各リンクに機械読み取り可能なアンダーライン80が付加されている。図5に示すように、これらは、ドキュメントのプリントプレビューバージョンに含めることもできる。同様のシステムが本願の譲受人に共通に譲渡された特許出願番号第08/372,784号で示されている。該出願は全ての目的のために引

用によってここに編入する。

【0083】付録Aに示すような、Webによって利用できる他の多くのエージェンシーが存在している。これら及び他のエージェンシー及びエージェンシーをトランスレーション（翻訳ソフト（translation））として使用することができる。

【0084】プリンタエージェンシーの形態の周辺エージェンシーは以上で説明された。入力装置用の周辺エージェンシーは、例えば、Webスキャナーとして用いることもできる。入力装置は、入力装置エージェンシーによって、デジタルスキャナー中で走査されるHTMLドキュメントのソースであると考えられる。ドキュメントのソースとして、Webスキャナーはクライアントに対しHTTPサーバとして現われる。ドキュメントはHTTP、GET、PUT及びPOSTコマンドを用いてソース化（sourced）される。進歩したWebスキャナーは要求に応じて提供するよう既に走査された全てのドキュメントのリストを管理することもできる。加えて、進歩したWebスキャナーはデジタルサインを付し、タイムスタンプを押しかつドキュメントを暗号化することができる。

【0085】サーチエージェンシーは利用可能なWebページのサーチを行うエージェントを有するWebエージェントである。Webのサーチツールは知られているが、しかし、それらは典型的にはサーバ上でローカルに実施するものであつて、サーバが見ることができるドキュメントは限られている。サーチエージェンシーによって、ユーザは任意のドキュメントセットのサーチを実施でき、かつ、サーチ結果をカスタマイズするか、又はフィルターエージェント又は他の処理エージェントに対しサーチ結果をユーザへ提示する前に扱わせることができる。

【0086】次に、図7は、トランザクションにアクション（act on）し、かつ、それを処理するプロセスを示すフローチャートである。このフローチャートにおいて、各ステップはS1からS8までのステップ数が付されている。プロセスの実行は、ステップS1でスタートし、かつ、他の指示する場合を除き数字順の増大に従って進行する。

【0087】ステップS1においては、一つのトランザクションがトランザクション列23から出される（図2、参照）。ステップS2では、リゾルバがインタレストドエージェントのリストを得るために整合したリストを走査する。前に述べたように、各エージェントはトランザクションの特徴の機能である基準（critaria）を有している。従って、各エージェントに対して、既存のトランザクションの特徴リストを用いて基準が評価され得る。もし、エージェントの基準に合致すると、該エージェントは、インタレストドエージェントのリストに付加される。

【0088】次に、リゾルバは各インタレストッドエージェントを呼び出す。より詳しくは、リゾルバは各エージェント（S3）に対するact_on()メソッドを呼び出す。以上で述べたように、act_on()メソッドは新規なトランザクションを追加し、既存のトランザクションを変更し、又は、トランザクションのハンドラ列にフックを付加することができる。各インタレストッドエージェントが一度トランザクション上で活動すると、ポインタはトランザクションのハンドラ列中に最初の見出し（entry）をセットする（S4）。最初のハンドラ列の見出しが現行のハンドラ列見出しであるので、現行のハンドラ列見出しによって、それに向けられたルーチン又はメソッドが実施される（S5）。トランザクションが満たされると（S6）、これは現行のハンドラが“true”の値を戻したとき表示され、もはや他のハンドラは呼び出されず、かつ、トランザクションは、トランザクションスタック（stack）から除去される。他方、もし、現行のハンドラ列見出しが“false”の値を戻したとすれば、そのときにはポインタは次のハンドラ見出しへ動かされる（S8）。トランザクションと協働するデフォルトハンドラ列見出しがハンドラ列の最後に設けられ、かつ、このデフォルトハンドラ列見出しによって指定されたプログラムコードは、常に、“true”の値を戻すので、トランザクションはその満足が保証される。典型的には、デフォルトハンドラ列見出しはエラーメッセージルーチン又はそれと同等のものである。勿論、少なくとも一つのハンドラ列見出しルーチンの連続した完結を保証しない代替実施例は存在し得る。この実施例において、ハンドラ列の消滅後に不満足なトランザクションを処理する代替方法は存在しないであろう。

【0089】以上の説明は、例示のためのものであって、限定のためのものではない。本発明の多くの変形例は、この開示を吟味することにより、当業者には明らかとなろう。従って、本発明の範囲は、前記説明を参照して決定すべきでなく、その代わりに、付記する請求項を均等の全範囲と共に参照して決定すべきである。

【0090】次に、本発明の理解をより容易にするために、追加の説明を行う。

1. 基盤施設（インフラストラクチャ）

A. エージェント・テクノロジー：

1) ベーシック・エージェント：ベーシック・エージェントは、1つの情報機器をドキュメントソース、ドキュメントトランスフォーマットおよびドキュメントコンシューマとして機能させる、DOCS（ドキュメント用計算機システム）の重要な構成品の1つである。

【0091】最初の試作エージェントのプロトタイプは、CERN httpサーバをベースとして使用して構築し、エージェントをPERLに書き込んだ。第2の試作エージェントは、PERLまたはJava内に構築した。これにより、特別目的コード（エージェント）をエ

ージェントの内部に直接ロードすることができ、効率がよく変更が容易になる。2つの実験エージェント、即ち、perlwwwライブラリとJigsawJavaサーバを構築したことは実際最良の選択であったろう。

【0092】エージェントを走らせるには、幾つかの異なる方法があり、複数のエージェントを有することにより下記の利点が生じる。

* エージェントとしてのエージェント：PERLまたはJavaのいずれかにエージェントを書きことができるので、複数のエージェントを持つことをこの場合有効である。

* エージェント内部のコード断片としてのエージェント：複数のエージェントを有することにより、PERLまたはJavaのいずれかを使用することができる。

* CGI（共通出入口のインターフェース）スクリプトとしてのエージェント：ドキュメント上で複数のエージェントを走らせてドキュメント自体を変更させたいという願望がより複雑なCGIスクリプトと成り得る。

* フィルタとしてのエージェント：これにより、パイプを発進しドキュメントを送送することができるエージェントを要求したりまたはそれを通じて要求することができる。

* 観覧者としてのエージェント：エージェントは、コマンドライン上でドキュメントを含むファイルの参照を許可されるプログラムである。

【0093】2) 物理的なエージェント（WebWart）：WebWartの名称は、〈Wall Wart（壁のこぶ）〉に由来し、壁のソケットに差し込む給電プラグを意味する。公式名称は、〈Peripheral Interface Agency（周辺インターフェースエージェント）〉または〈Web Peripheral Interface（ウェブ周辺インターフェース）〉等である。このWebWartは我々にエージェント制に基づき周辺機器を納得のいくように実演させることを許容する。WebWartは我々が周辺機器を取り付けるエージェントの物理的な実施態様である。

【0094】最も簡単な場合、試作WebWartはディスプレイのない小さい箱に入ったPC（パーソナルコンピュータ）であり、価格は、RAM（ランダムアクセスメモリ）と計算能力によるが500ドルから1000ドル程度である。これらの製品は、IPアドレスとホスト名等をHTTPを介して設定できるようにする小規模なソフトウェアを有し、PCと区別するために独特な色に塗装する。可搬式WebWartは、若干高価で、リコー（日本）よりラップトップ又はパームトップ型機器としてディスプレイなしで供給される。形状及び目標価格範囲は、プリンタ用追加ネットワークインターフェースのような形状で、300ドルから500ドルの範囲内の価格である。

【0095】最終的には、WebWartは、現実の製品として、PCMCIAカード（Webカード）からラックマ

ウントに至る広範な寸法の製品であり得る。理由は2つある。

1. 従来の周辺機器とつないで使用する。
2. 実験室機器のような特別目的の周辺機器と接続して使用する。
3. WebWart製品は、PIAおよびWebファイルの物理的な実施態様である。
- 3) 縮尺可能なエージェンシー：最小レベルの縮尺可能なエージェンシーは、カメラ、キーボード、ページャ（マジックペンも同様）と同じ程度に小型なネットワーク装置に組み込むことができる非常に軽量のエージェンシーで有り得る。シングルスレッド式であり、簡単なプロとコールを使用し、MIMEタイプの制限セットを使用することができる。これは、通常、小型固定式エージェンシーにのみに接続する。

【0096】次のレベルは、通常、PIAとして働く中間重量エージェンシーであり得る。これは、マルチスレッド式で、ディスク、ネットワーク及び広く種々の言語で書かれたエージェンシーに接続できる。

【0097】最高のレベルの縮尺可能なエージェンシーは、ペンティアム搭載パソコンからイカルスまでの範囲のハードウェアで稼働できる、アパッチ（Apache）スケールの大サーバエージェンシーである。

【0098】1つのエージェンシーは、単一プログラムというよりは、アーキテクチャであり、C、C++、Perl及びJavaを含む種々の言語でのエージェンシーを生み出すことが可能でなければならない。エージェンシーが共通のコードベースを共有することができない場合でも、（IDLで規定できる）インターフェース及び／または各クラスの共通セットを実行可能なように共有できる。

【0099】埋め込みエージェンシー：WebWidgets、WebWindows及びWebCardsのようなものを可能にする技術は、小さな（tiny）埋め込み可能なエージェンシーである。埋め込み可能なエージェンシーは、電力消費量が小さく、簡単な近距離無線インターフェースを有し、広範囲の単一ハードウェアに接続することが必要である。

【0100】プロトタイプの場合、データ（ドキュメント）と同時にコード（エージェント）をロードする能力を有する必要がある。これは、機密保持のために何らかの方策を実施せねばならないことを意味している。エージェントをドキュメントとしてパッケージすることにより、インストールの方法を容易にする。

【0101】16ビットのコントロールチップシリーズの1つを、特に、エージェントが少なくとも1メガバイトの各コード及びデータにアクセスする能力を有している場合に使用する。エージェンシーを128Kに圧縮することは難しいが、可能とすべきである（WebFile WebWalletは、Linuxと標準PIAアーキテクチャを用いてインテル社のチップに実装することが可能である。イン

テルのラップトップ型CPUモジュールを用い、WebWalletのプロトタイプとすることも可能であろう）。

【0102】B. ドキュメント指向の計算：

1) インターフォーム（InterForm）：インターフォーム（インターフェース・フォーム）は、プログラムをHTMLのユーザ・インターフェースのフォームと、ユーザレベルドキュメントと実行レベルドキュメントの双方と結合させる単一ドキュメントである。

* 埋め込みコードフラグメント、例えば、〈コード言語=PERLインターフェース=CGI〉を持つHTMLドキュメント。

* オンザフライでHTMLを生成する適当なツールキットを持つ種々の言語でのコード。

* 適当なMakefileを持つ混成HTML及びコードを内容とするディレクトリ、かようなディレクトリは、tarファイルに束ね、適当なCGIインターフェースを用いHTTPを介して送り出すことができる。取り込みすべて適当なインデックス.htmファイルである。“クライアント”端では、エージェンシーのファイル内の正しい場所にかようなパッケージをインストールするための機構を必要とする。“インストール・エージェンシー”はこのために使用する。

【0103】2) DOFSプロトタイプ：エージェンシーは、恣意的な名前の属性をドキュメントと結合させるために拡張したドキュメントのキャッシュ記憶装置を必要とする。このキャッシュ（cache）記憶装置は、WebFileのためのプロトタイプ・ファイルシステムとして使用できる。

【0104】最良の実施例の場合、各ドキュメントは、原ドキュメントと任意の要求追加ドキュメントの両方を含む1つのディレクトリにより代表されるようにする。1つの結果として、キャッシュをWebブラウザで表示する際に、原URLと全ての随伴情報の間に簡単明瞭なマッピングを存在させる。ドキュメントの構文解析ツリー表現を使用することができる。その場合、マップファイルは、各記憶ドキュメントを伴う。

【0105】3) ドキュメント指向ファイルシステム（DOFS）：より長期には、テキストを含むファイルシステムから構文解析ツリーを含むファイルに移行させたと考える。本来、テキストは、常に、以前から受け継がれてきたアプリケーションによって使用でき、ネットワークで送信される。

【0106】これには2つのバージョンがある。

1. Unixファイルシステムに保管されるように設計したDOFS。

2. 専用ディスク、区分ディスクまたはファイルに保管されるように設計したDOFS。

【0107】DOSファイルシステム上でDOFSを働かせることは、制限が大き過ぎるので、実際的ではないであろう。しかしながら、専用区分バージョンは、ウィ

ンドウズのスワップファイルのように、大きなDOSファイルに容易に保管できた。動的ファイルを持つために、任意レベルのノードをコードで置き換えられるようにすることが必要である（これは、非常に〈Self〉に似ている）。

【0108】表示については幾つかの可能性がある：

1. マップファイルを伴うテキストファイル。これは、
 - ・生テキストを検索するには最速で、
 - ・非常にコンパクトであり、
 - ・シークを用い非常に大きなファイルを取り扱うことができ、
 - ・バッファされない限り低速ランダムアクセスとなる。
2. 前記ノードにおいて、2値である直線化した構文解析ツリー。これは、
 - ・構文解析ツリーをメモリに格納するには最速であり、
 - ・バッファされないランダムアクセスは最低速である。
3. OODB：
 - ・スパースランダムアクセスの場合早く、
 - ・順次、アクセスは最低速で、
 - ・コードヒットが大きい、
 - ・高価である。
4. DOFS（ドキュメント指向ファイルシステム）：
 - ・徹底したカスタムデザインで、
 - ・特に、構文解析ツリー向きのファイルシステムで、
 - ・httpからワードに様にアドレスで、
 - ・小さなファイルを持つ従来のファイルシステムと見なすことができ、
 - ・…又は〈凍結〉リストの処理機またはSelfに似たバーチャルメモリと見なすことができる。

【0109】テキスト+マップのオプションは、最初のよき妥協解であるように見え、下記の利点が追加される：

- * マップはプラットフォーム依存2値式とすることができ、
- * 従来のソウトにはインパクトを与えず、
- * テキストからマップを容易に再構築でき、
- * そのファイルが変更できなければ、放棄することができ、
- * ファイルとマップを別々に圧縮することができる。

【0110】前記ドキュメント指向ファイルシステムは、現存“WWFS”に類似のインターフェースを有し、このインターフェースにより前記Webにそれがファイルシステムであるかのようにアクセスすることができる。例えば、選択子の構文（Syntax）が実働化（implementation）に依存するドキュメント範囲において、〈URL##selector〉を使用してアドレス指定をすることができる。

【0111】4）ドキュメント指向プログラミング言語ドキュメント指向プログラミング言語は、下記の2通りの異なる方法でドキュメント指向にする。

1. SGMLのようなシンタックスを持ち、ドキュメントに容易に埋め込めるようにする。

2. ドキュメント上で直接働く、即ち、ドキュメントが第1類値（first-classvalue）となる。この言語は、LISPまたはSelfにかなり似ている。即ち、LISP（リスト処理言語）のようにプログラムとデータが同一の表現を有しSelfのようにデータが属性値のリスト構造を有する。

【0112】C. エージェント・テクノロジー：エージェントは勿論、Interformsとして実現でき、ドキュメント指向ファイルシステムに記憶できる。本節では、特殊エージェントについて説明する。

【0113】1）WebWidgetインターフェース：WebWidgetはWeb使用可能な入力装置である。これはクライアントに似ており、何らかのイベントが発生した時に予め定められたURLにドキュメントを転記（POST）するように設計されている。WebWidgetは下記を含んでいる。

- * 押しボタン（シングルビットを入力する）。
- * キーボード（シングルキャラクターを入力する）。
- * キーボード+1ラインディスプレイ（文字列を入力）。
- * タッチスクリーン（座標を入力）。
- * カメラ/スキャナ（イメージ及びそのパラメータを入力）。
- * マイクロフォン（音声流を入力）。
- * ビデオカメラ（ビデオ+オーディオを入力）。
- * GPS受信器（座標を入力）。

【0114】2）WebPrinting：Webプリンタおよび幾つかのWebウインドウズのようなページ装置を働かせるために、HTMLレンダリングは、下記のようにアドレスを指定する必要がある。

- * ページ付けとページレイアウト。
- * スタイルシートの適用。
- * 高速、低速レンダリング。
- * マルチファイルドキュメントの平面化。
- * ポストスクリプト、アクロバット及びその他の専有のフォーマットを埋め込みイメージまたは然るべきMIMEタイプ付き全ドキュメントのいずれかとして処理する。

* オプション（InterFormのデザイン）の指定。

【0115】3）オブジェクト・インターフェース：これは、CORBA及びOLEオブジェクトに対するインターフェースを提供する。このオブジェクト・インターフェースは、〈ライブドキュメント〉と対応するサーバ側コード（必要ならばクライアント側コード）を自動的に構成し、任意のオブジェクト・インターフェース名を与える。

【0116】この問題に関しては、実際に幾つかのアスペクトがある。

- * 実行時間インターフェースを使用し、オンザフライ

式印刷のフォームを構成し取り扱う。

* IDL記述を前処理して当該フォームを翻訳 (Compile) する。

* ドキュメント指向インターフェースフォーム (InterForm) の総称マークアップをオブジェクト指向オープンドキュメント (OpenDoc) 部分とコンテナに併合する。

【0117】D. その他のテクノロジー:

1) BodyWeb: 多くのアプリケーション (例えば, WebWallet, WebCard, WebPAD) は、小電力、短距離 (1メートル半径)、高帯域幅 (100メガビット/秒) のローカルネットワークから恩恵を受ける。かようなネットワークは、携帯または座席の近辺で使う物にはすべて接続できる。使用できる2つの技術は、(ゼロックスが偏在計算プロジェクトで使用している) 誘導結合であり、また、無線 (radio) である。赤外線は見通し線の問題があるので、ポケットに入れることができない。超音波は帯域幅と干渉の問題がある。スペクトル拡散無線がベストであろう。

【0118】2) ライブドキュメント: ライブドキュメントは、発行数を高める。

* ライブドキュメントのフレームワークの構成方法は?

* ライブドキュメントを作成するのに必要な道具 (tools) は?

* 例えば、CORBA ILDの仕様書からライブドキュメントを自動構成できるか?

* ユーザに提供しなければならないツールは? 編集及びカスタマイズするためには任意のテキスト・エディタを用いることができるが、幾つかの選択商品から適したものを選択できる筈である。

* ライブドキュメントを働かせるために、サーバ側に必要なDGIは?

【0119】2. アプリケーション:

A. エージェント:

1) 個人情報エージェントとエージェント: 個人情報エージェント (PIA) は、他の場所で説明するが、構成用の僅かなインターフェース・フォーム (Interforms) とキャッシング、注釈、一時記憶装置及びエージェントの記憶装置に用いられるドキュメント指向ファイルシステム (DOFS) を有する簡単なエージェントである。

【0120】2) PIAエージェント: エージェントも他の場所で説明するが、殆どのエージェントは簡単で、その多くは、ワン・オフ・ハック (One-off hacks) のように、特別に書かれ、そのURLを回って交換される。

【0121】3) WebFile: Webファイルは、専用ドキュメントサーバである。なお、バージョンコントロールのためにCVSを使用する選択権をユーザに付与しなければならない。また、機密保護のための暗号を含まねば

ならない。好ましくは、全ての構成を、Webを使用しに行くことである。CVSリモートアクセスサーバを変更し、Rshの代わりにHTTPを使用するようにすれば有益である。ドキュメントをWebファイルに取り込む好ましい方法は、代理権を与えて、Webファイルをキャッシュとして有効に使用することである。

【0122】プロトタイプWebファイルは、大ディスクを持つWebWartである。プロトタイプ用ソフトウェアは、若干ありふれたものではない。以下のことを実行する必要がある。

* ネットワークの管理 (IPアドレス、マスク、ゲートウェイ、ノード名等)。

* ファイルの操作 (PUT要求、URLからのコピー、改名、削除等)。

* リンクの保守 (ファイルを移動させる時のリンクの改名等)。

* バージョンコントロール (Jigsaw WebサーバはすでにCVSをサポートを有している)。

【0123】4) WebCD-R: WebCD-Rは、WebファアルとWebプリンタの間のある交点である。全てのWebサイトを含むドキュメントをその中にロードし代理権を与える。ドキュメントは、次に、CD-Rに書き込むことができる。CD-Rの内容は、本質的に代理キャッシュのイメージでなければならない。CD-Rは、次に、代理サーバを使用して走査され、キャッシュの日付と現在の日付と比較し、ネットワークから変更されたページを得る。

【0124】ソフトウェアの必要条件は、プリンタまたはファイルサーバのいずれの場合よりもかなり簡単であり、その理由は次の通りである。

* ファイル操作の必要がない。

* バージョンコントロールの必要が全くない。

* プリンタの場合よりかなり“スタイル”パラメータが少ない。

* mk iソフトが既に存在する。

一方、代表的な実施例の場合、ソースツリーと宛先.isoファイル用に少なくとも1500メガバイトのディスクを必要とする。

【0125】5) WebCard: これは、近距離高帯域幅無線または有線ネットワークへのリンク付PCMCIAサイズのエージェントであり、デジタルカメラ及び掌上型のような既存PCMCIA装置をWeb装置として機能させることができる。このWebカードはまた小さなポータブル型のWebファイルとして独立して機能でき、例えば、感知可能データ、デジタルキャッシュまたは個人の識別用のカードとして機能できる。1つの可能な実施例の場合、PCMCIAのII型フォームファクタとコネクタを有し、フラッシュメモリと充電可能なバッテリを付属させている。また、前記PCMCIAコネクタより低い帯域を有するIrDAまたは誘導リンクを持

つこともでき、これはスロットとコネクタを必要としない。Webカードは、より多くのパワーを与えるための“マイクロダグ”と、イーサネット接続用のスロットとモデム、ページャ、オーディオ、カメラ又はディスク用のスロットを有することが望ましい。

【0126】6) WebWallet

これは、Webファイルの“ノートブック”バージョンである。小形であるが、PCMCIAサイズではない。実際には、1個のPCMCIAスロットを持つことが好ましい。軽量でバッテリー式でなければならない。シングル・ユーザであれば、PIAとしても機能する。

【0127】特別仕様として、下記を有することができよう。

- * 2.5インチ1ギガバイト(予想)の内蔵ディスク。これは、ユーザが完全に個人の場合には十分である。
- * 充電式バッテリー。バックライト式ディスプレイがなく、大電力のCPUが不要であり、ディスクを連続運転しないので、バッテリーの寿命は良好である。セルラ電話のバッテリーを使用するのも良い選択である。
- * PCMCIAのII型スロット。このスロットは、WebPad, RDC-1フラッシュカード、イーサネット、モデム、カメラ等に使用される。
- * IrDA誘導リンク。
- * オーディオ入出力用コネクタ。
- * 高速デジタル入出力装置(多分Fireire)。
- * モデムまたはリコーネット用直列ポート。
- * ベルトフックまたはパウチ。WebWalletはカメラまたはパーソナルステレオパウチに挿入できなければならない。

【0128】WebWalletは、ネットワークへの接続に使用できる小形で安価な充電器/ドッキングステーションを必要とする。これは、携帯電話用ベースステーションというよりは受け台であり、または、ユーザ側で何らかの作業または手探りを要するものではなくて、パイロットPDAの受け台であることが必要である。このアイデアは、ユーザがノートブックまたはハンドバックを持って行くのと同じように、WebWalletをどこにでも持ち運べるようにするためである。近距離無線リンクが十分な帯域幅を有している場合は、夜間を除いて受け台に置くことも必要としない。

【0129】使用には、下記を含む。

- * WebCard及びWebPAD用ドック充電器。
- * RDC-1写真アルバム。
- * デジタルスクラップブック。
- * ノートブック(Webウィンドウズ付)。
- * PAD用バックアップ。
- * デジタルカメラ(PCMCIAカメラ付)。
- * CD品質オーディオレコーダ/プレーヤ。
- * 電話(受話器とセルラ又はリコーネットモデム

付)。

【0130】写真アルバムであることに加え、WebWalletは、PDAに用いられているような他のPCMCIA 1モリカードと一緒に使用できる。WebWalletは、ユーザのドキュメント階層構造中に配置させる前記カードボリュームIDを使用することができ、ロード、アンロードまたは前記カード上のファイルと同期するようにプリセットすることができる。

【0131】B. 周辺機器:

- 1) WebPrinter: このWebプリンタは、
 - * HTMLを直接印字し、
 - * それ自身のHTTPの要求をなし、
 - * シームレス走査検索、印字およびプレビュー用プロキシとして使用でき、
 - * 印刷またはプレビューするドキュメントの問い合わせ列を与えられるCGIを有する。

【0132】ハードウェアのプロトタイプは、リコーのレーザプリンタとWebWartを用いて構築することができた。各プリンタのホームページへ導くクリック駆動式マップを設けることができる。最終記録 (Last-minute) の走査及びフォーマットの引き出し (tweaking) のために各Webプリンタの次にWebウィンドウズを持つと有益である。(コントロールパネル) 上に幾つかのプリンタソフトを置くことにより、実際に上手く工夫 (cheat) できた。勿論、最終的には、MFMのスキナ、プリンタ、ファクシミリ、複写機、IM3等のWebで実行可能な完全セットを持ちたいと望んでいる。

【0133】2) WebWindows: 実際のWebWindowsは、ブラウザではない! それは、完全に成熟した情報機器であり、基本的にはサーバのようなものであり、PIAと密接に稼働するように設計されている。従って、従来のオラクル/サン/アップル/のどのネットワークコンピュータとも全く異なるものである。

【0134】Webウィンドウズは、PIAとのみ稼働するように設計されている。かように、その(ブラウザ)のアスペクトは、最小限に削減されている。PIAはWebウィンドウズと、GET要求に回答するだけでなく、PUT要求を送ることにより、相互作用する。Webウィンドウズは、かなりの程度、Xサーバに似たディスプレイサーバまたはWebプリンタとして機能する。この結果、Webウィンドウズは、オプション・メニューまたは他の構成情報を必要としない。即ち、連携PIAからすべてを取って来る。Webウィンドウズを任意の人数のユーザで使用することも容易である。即ち、ユーザはPIAにログ・インするだけでよい。

【0135】ハードウェアとして、独立型Webウィンドウズを、通常、WebWalletまたはPIAを走らせるPCと共に使用する。その他の機器、例えば、Webプリンタと結合したWebウィンドウズは、現在のユーザのWebWalletを使用し、識別と状態情報を得ることができ

る。Webウィンドウズは、ネットワーク・コンピュータよりもディスクの必要度は低い。Webウィンドウズは、サーバをすでに内蔵しているので、PIAと容易に一体化することができる。設計書は、Webウィンドウズのための要求事項の規定と最良の使用法の図解を含んでいる。

【0136】3) WebWindows Mock-up: Webウィンドウズのモックアップは、Webウィンドウズではない。Webウィンドウズ・モックアップは単純であり、GI200またはWebブラウザを実行するシンクパッド(ThinkPad)である。大衆を説得するために役立つので重要である。GI200はリコーにより製造されている事実が説得を助けている。

【0137】追加作業を必要とするのは、下記の通りである。

- * GI200にLinuxを設置する(但し、PIAを稼働させたい場合)。
- * ブラウザのPIAに接続する。
- * 2つのウィンドウズを有効に利用する。即ち、1つを作業に(Pushing to)に、もう1つを検索に使用する。
- * GI200を他の機器(例えば、プリンタ)にコントロールパネルとして取り付けることができるように、エレガントな(物理的)支持ブラケットを作製する。

【0138】残念ながら、GI200はかなり高価であるので、他の可能性を考慮する必要がある。PC-110はその1つである。即ち、Linuxを実行できることは知られており、(ある場合には有効な)キーボードを有している。新ザウルスMI-10がもう1つの可能性である。これは、明らかに66MHz 486カラーディスプレイ、ウィンドウズ95及びWebブラウザを有している。

【0139】4) WebImager: Webイメージャは、イメージ用のWeb入力装置である。この装置は下記を実行しなければならない。

- * 指定URLにイメージを置くかまたは記入する。
 - * イメージの移動トラックを維持する。
 - * 環境設定用エージェンシーを使用する。
- 当然、これはWebCam、RDC-1及びIM3と密接に関連する。Webイメージャは、フラットベッド式スキャナ、ファックカモデム、コネクチック・カメラまたはRDC-1フラッシュカードを入力装置として使用できる。ユーザ・インターフェース・ソフトウェアはいずれの場合も同じである。低レベル・キャプチャ(capture)ソフトのみ異なる。

【0140】5) WebPAD: これは、ポケットサイズのシングルユーザ用Webウィンドウズである。これは、研究用というより物理的なプロトタイプのパッケージデザイン用のものである。実際の研究は、その使い方を研究することから始まる。WebPADによる実験方法は、CR

Cにおいて、各人に1個のWebPADを装備、何が開発されるかを見る。USRのパイロットPDAを使用してプロトタイプ化することも可能である。特別に、下記を想定する。

- * 3x5インチを越えないサイズ。
- * フラッシュメモリ。
- * IrDAまたは誘導リンク。
- * タッチ式液晶ディスプレイ、小形スピーカ。
- * 再充電可能なバッテリー。

10 【0141】PCMCIAコネクタは、ラップトップ、マイクロ・ドックまたは高速接続用WebWalletに挿入することができる。マイクロ・ドックは電力を上げ、イーサネット接続用スロット、モデム、ページャ、オーディオ、カメラまたはディスク用のスロットを提供する。液晶ディスプレイは、下記の幾つかの機能を有している。

* デフォルト: ディスプレイ“ビジネスカード”。

* 機密保持のために署名を用いる。

* 交換データ量をコントロールする。

* 制限データエントリのようなPDAのアスペクト。

20 WebPAD-WebCardの組み合わせは、下記のように機能できる。

- * デジタル財布/小切手帳/クレジットカード。
- * ビジネスカードホルダー(日本の慣習である名刺交換に最適)。
- * 名札/バッジ/キー。
- * 個人データ(名前、住所、病歴等)の保管。
- * 時計/カレンダー。
- * 計算機。

30 【0142】6) その他の機器: その他の機器には、WebScanner, WebCam, WebFAX及びWebMFMが含まれる。WebMFMは完全にWebで実行可能で、IM3、プリンタ、スキャナ、ファクシミリ等と組み合わせて一体化できる。さらなる研究により、印刷ドキュメントの書式を、そのドキュメントを入力装置(ライブペーパー)として使用できるようなフォーマットする方法を知る。WebFAXの技法はかなりの処理を要する。処理を軽減するためには、下記を必要とする。

- * 全てのリンクをURLとドキュメントタイトル(既知の場合)及び最終に修正した日付を含む脚註としてフォーマットする。
- * 各脚註の隣にチェックボックスを置き、脚註番号用の小さなバーコードをつける(オプションとして、これらのチェックボックスを行にフォーマットして脚註を削除するかあるいは全ての脚註を最後の単独頁に置くことが可能である)。
- * 2Dのバーコードを、ヘッダ、フッタまたはマージンに置き、開始脚註番号と(1)ページの完全なURLとタイムスタンプまたは(2)プリンタのURLとハッシュコードのいずれかを付ける。

50 【図面の簡単な説明】

41

【図1】 ドキュメント指向コンピュータシステムにおいて、色々なWebクライアントとサーバ間に挿入された本発明によるエージェンシーのブロック図である。

【図2】 単一のWebエージェンシーとその内部構造のブロック図である。

【図3】 Webエージェンシーを通してWebクライアントからWebサーバへのドキュメントの要求のサンプル要求フローのブロック図である。

【図4】 図3の要求フローよりも複雑なサンプル要求フローのブロック図である。

10

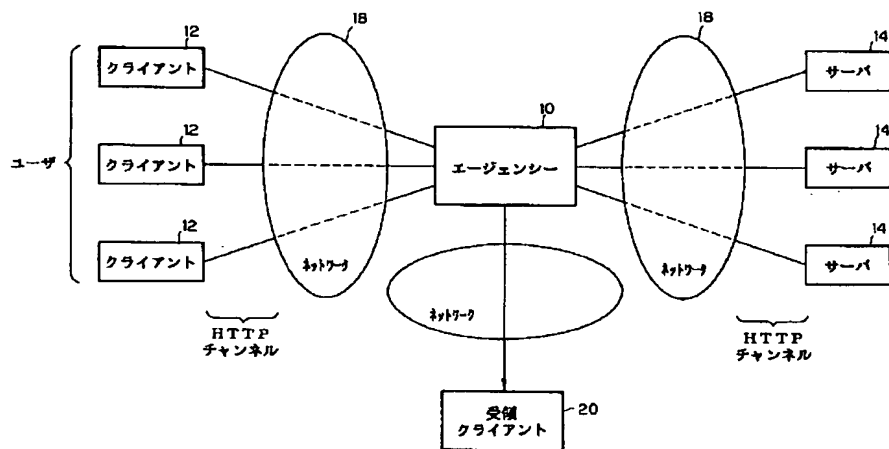
42

【図5】 Webプリンタエージェンシーに合わせて環境設定されたWebエージェンシーのブロック図である。

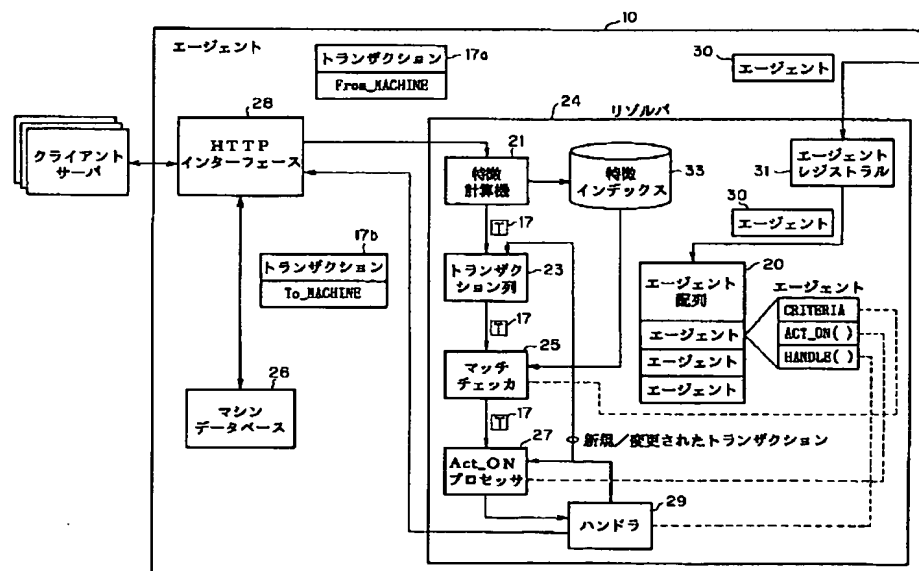
【図6】 本発明によるWebプリンタエージェンシーを用いて好適に印刷された、機械読み取り可能なアンカーリンクで印刷されたハイパーテキストページを図示したものである。

【図7】 トランザクションに作用しかつ処理する方法を図示したフロー図である。

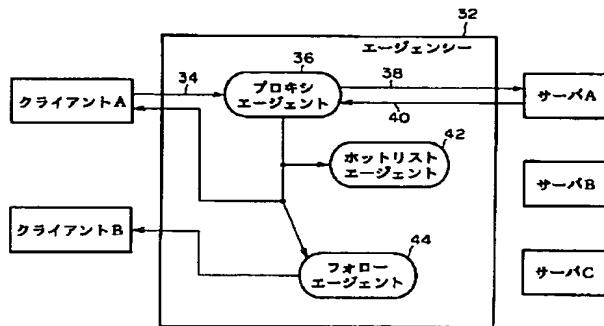
【図1】



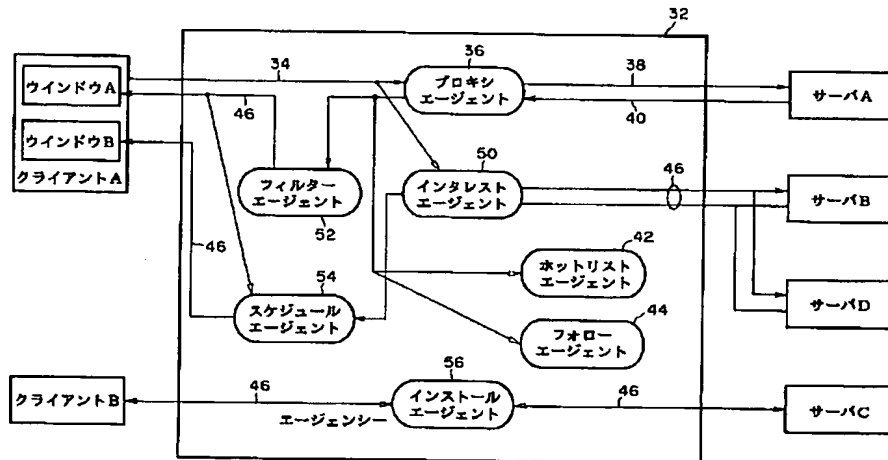
【図2】



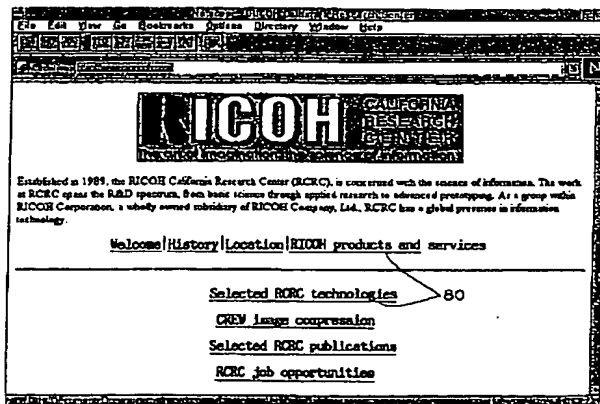
【図3】



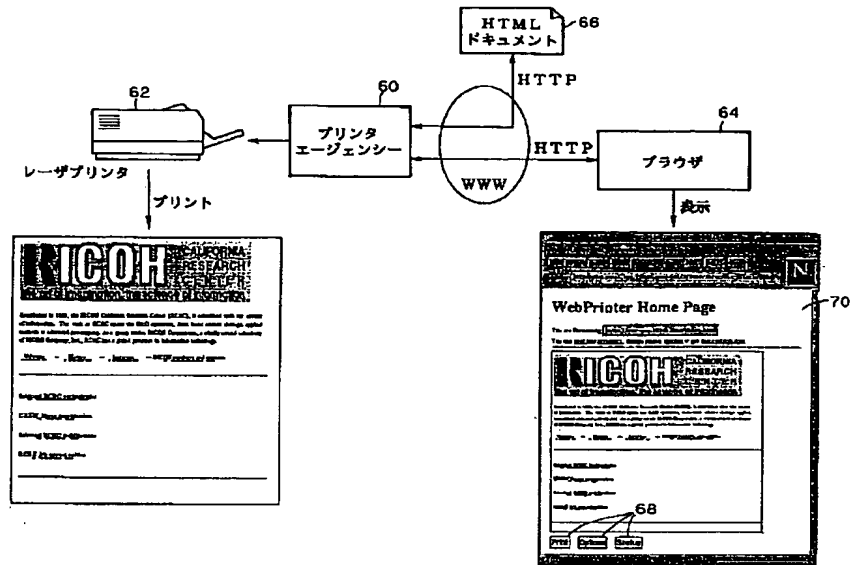
【図4】



【図6】



【図5】



【図7】

